

**DESARROLLO DE UN PROTOTIPO DE UN SISTEMA DE INFORMACIÓN
PARA FINCAS PRODUCTORAS DE CAFÉ**

DANIEL TORRES BEDOYA

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, FÍSICA Y
CIENCIAS DE LA COMPUTACIÓN
PROGRAMA INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PEREIRA**

2016

**DESARROLLO DE UN PROTOTIPO DE UN SISTEMA DE INFORMACIÓN
PARA FINCAS PRODUCTORAS DE CAFÉ**

DANIEL TORRES BEDOYA

**PROYECTO DE GRADO PARA OPTAR AL TÍTULO DE
INGENIERO DE SISTEMAS Y COMPUTACIÓN**

**LUZ ESTELA VALENCIA AYALA
INGENIERA INDUSTRIAL
ASESOR DE PROYECTO DE GRADO**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, FÍSICA Y
CIENCIAS DE LA COMPUTACIÓN
PROGRAMA INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PEREIRA**

2016

Tabla de contenido

1. INTRODUCCIÓN.....	6
2. DESCRIPCIÓN DEL PROBLEMA.....	7
3. OBJETIVO GENERAL	8
4. OBJETIVOS ESPECÍFICOS	8
5. ALCANCE.....	8
6. MARCO TEÓRICO	9
6.1 INGENIERÍA DE SOFTWARE.....	9
6.2 ¿QUÉ ES LA INGENIERÍA DE SOFTWARE?.....	9
6.3 ¿QUE ES EL PROCESO SE SOFTWARE?	10
6.4 METODOLOGÍAS AGILES	11
6.4.1 METODOLOGÍA SCRUM.....	11
6.5 LEVANTAMIENTO Y ANÁLISIS DE REQUERIMIENTOS	13
6.5.1 HISTORIAS DE USUARIO.....	14
6.6 DISEÑO DE SOFTWARE	15
6.6.1 LENGUAJE UNIFICADO DE MODELADO UML	15
6.6.2 VISTAS 4+1	16
6.7 ARQUITECTURA DE SOFTWARE.....	17
6.7.1 MODELO VISTA CONTROLADOR.....	17
6.8 CODIFICACIÓN	18
6.9 ESTÁNDARES WEB.....	19
6.9.1 HTML (HYPER TEXT MARKUP LANGUAGE)	19
6.9.2 XML (EXTENSIBLE MARKUP LANGUAGE)	19
6.9.3 JAVASCRIPT.....	19
6.9.4 CSS (CASCADING STYLE SHEETS)	20
6.9.5 PHP (PHP: HYPERTEXT PREPROCESSOR).....	20
6.9.6 FRAMEWORK	21
6.9.7 CODEIGNITER	21
6.9.8 MY SQL	21
7. DEFINICIÓN DE LOS REQUERIMIENTOS DEL SISTEMA	22
7.1 HISTORIAS DE USUARIO REQUERIMIENTOS NO FUNCIONALES	22
7.2 SPRINT 1	23
7.2.1 HISTORIAS DE USUARIO.....	23

7.2.2 VISTA DE ESCENARIOS (CASOS DE USO)	27
7.2.3 VISTA DE PROCESOS (DIAGRAMAS DE SECUENCIA)	35
7.2.4 EJECUCIÓN DE CASOS DE PRUEBA.....	¡Error! Marcador no definido.
7.3 SPRINT 2	40
7.3.1 HISTORIAS DE USUARIO	40
7.3.2 VISTA DE ESCENARIOS (CASOS DE USO)	42
7.3.3 VISTA DE PROCESOS (DIAGRAMAS DE SECUENCIA)	46
7.3.4 EJECUCIÓN DE CASOS DE PRUEBA.....	¡Error! Marcador no definido.
7.4 SPRINT 3	50
7.4.1 HISTORIAS DE USUARIO	50
7.4.2 VISTA DE ESCENARIOS (CASOS DE USO)	51
7.4.3 VISTA DE PROCESOS (DIAGRAMAS DE SECUENCIA)	54
7.5 VISTA LÓGICA (DIAGRAMA DE CLASES)	56
7.6 VISTA DE DESARROLLO (DIAGRAMA DE COMPONENTES)	56
7.8 VISTA FÍSICA (DIAGRAMA DE DESPLIEGUE)	57
7.9 MODELO RELACIONAL BASE DE DATOS	57
8. APLICATIVO	58
9. CONCLUSIONES	59
10. BIBLIOGRAFÍA	60
11. MANUAL TÉCNICO	601
12. PRODUCT BACKLOG	77

Tabla de ilustraciones

Ilustración 1 Proceso Scrum	12
Ilustración 2 Vistas 4+1	16
Ilustración 3 Diagrama de Caso de Uso: Usuario Registrado	27
Ilustración 4 Diagrama Caso de Uso: Dueño de Finca.....	30
Ilustración 5 Diagrama Caso de Uso: Administrador	32
Ilustración 6 Diagrama de Secuencia #1: Autenticación.....	35
Ilustración 7 Ilustración 7 Diagrama de Secuencia #2: Gestión de datos básicos.....	35
Ilustración 8 Diagrama de Secuencia #3: Recuperación de contraseña	36
Ilustración 9 Diagrama de Secuencia #4: Registro	37
Ilustración 10 Diagrama de Secuencia #5: Creación de perfil administrador	37
Ilustración 11 Diagrama de Secuencia #6: Crear perfil trabajador	38
Ilustración 12 Diagrama de Secuencia #7: Gestionar información del perfil trabajador	39
Ilustración 13 Diagrama de Caso de Uso: Dueño de Finca.....	42
Ilustración 14 Diagrama de Caso de Uso: Administrador	44
Ilustración 15 Diagrama de Secuencia #8 Gestión de finca.....	46

Ilustración 16 Diagrama de Secuencia #9 Gestión de lotes	47
Ilustración 17 Diagrama de Secuencia #10 Gestión de Cultivos	48
Ilustración 18 Diagrama de Secuencia #11 Gestión de Labores Culturales.....	49
Ilustración 19 Diagrama Caso de Uso: Producción	51
Ilustración 20 Diagrama de Secuencia # 12 Registrar Producción	54
Ilustración 21 Diagrama de Secuencia #13 Visualizar Reportes.....	55
Ilustración 22 Diagrama de Secuencia #14 Visualizar Pagos	55
Ilustración 23 Vista Logica.....	56
Ilustración 24 Vista de Desarrollo	56
Ilustración 25 Vista Física	57
Ilustración 26 Base de Datos	57

Lista de Tablas

Tabla 1 Historia de Usuario 1	22
Tabla 2 Historia de Usuario 2	22
Tabla 3 Historia de Usuario 3	22
Tabla 4 Historia de Usuario 4	23
Tabla 5 Historia de Usuario 5	23
Tabla 6 Historia de Usuario 6	24
Tabla 7 Historia de Usuario 7	24
Tabla 8 Historia de Usuario 8	25
Tabla 9 Historia de Usuario 9	25
Tabla 10 Historia de Usuario 10	25
Tabla 11 Historia de Usuario 11	26
Tabla 12 Historia de Usuario 12	26
Tabla 13 Caso de Uso 1	27
Tabla 14 Caso de Uso 2	28
Tabla 15 Caso de Uso 3	29
Tabla 16 Caso de Uso 4	30
Tabla 17 Caso de Uso 5	31
Tabla 18 Caso de Uso 6	33
Tabla 19 Caso de Uso 7	34
Tabla 20 Historia de Usuario 13	40
Tabla 21 Historia de Usuario 14	40
Tabla 22 Historia de Usuario 15	41
Tabla 23 Historia de Usuario 16	41
Tabla 24 Caso de Uso 8	42
Tabla 25 Caso de Uso 9	43
Tabla 26 Caso de Uso 10	44
Tabla 27 Caso de Uso 11	45
Tabla 28 historia de Usuario 17.....	50
Tabla 29 Historia de Usuario 18	50
Tabla 30 Historia de Usuario 19	50
Tabla 31 Caso de Uso 12	51
Tabla 32 Caso de Uso 13	52

1. INTRODUCCIÓN

Desde hace más de 50 años la producción de café es una de las actividades clave para la economía colombiana ya que exporta más de 12 millones de sacos al año¹; a pesar de esto la administración de los cafetales se realiza de forma manual ocasionando que no se aprovechen al máximo los elementos que los caficultores disponen a la hora de realizar mantenimiento a sus cultivos.

La adecuada administración de las fincas permite que se tomen mejores decisiones respecto los insumos, trabajadores y capital necesario para el sostenimiento del cultivo del café; por tanto el desarrollo de un software constituye una ayuda a los dueños y administradores de las fincas en la correcta realización de esta administración, mediante la sistematización de procesos como registro de empleados frecuentes y eventuales, nómina, labores de mantenimiento del cultivo y análisis de rentabilidad del cultivo.

La sistematización de estos procesos permite la retroalimentación del sistema de planificación, seguimiento y evaluación de los procesos ya que existe la posibilidad de realizar comparaciones con datos históricos de la finca y en particular de cada uno de los lotes asociados a esta, dando la posibilidad observar el comportamiento de la finca en diferentes periodos de tiempo y así tomar decisiones respecto a su situación.

Para lograr la sistematización de estos procesos se recurre a técnicas y metodologías de ingeniería de software que permitan realizar un desarrollo progresivo del mismo como lo son las metodologías ágiles, para la realización de este proyecto se utilizó la metodología SCRUM y el modelo Vista Controlador los cuales permiten realizar un análisis, diseño, codificación y pruebas del producto de forma incremental, asegurando que se maximice la flexibilidad y adaptación respecto a las necesidades del cliente, y se disminuya el tiempo de codificación y los errores cometidos durante su ejecución.

¹ <https://www.federaciondecafeteros.org/static/files/Caldas4.pdf>

2. DESCRIPCIÓN DEL PROBLEMA

Colombia es uno de los principales productores de café en el mundo, y teniendo en cuenta que el departamento de Caldas es el cuarto mayor productor de café del país aportando aproximadamente 9 millones de arrobas de café pergamino seco al año con 87.127 hectáreas sembradas en café es claro que una de las actividades clave para el desarrollo del departamento es la producción de café, por tanto es de vital importancia que los procesos que se llevan a cabo en las fincas cafeteras se realicen de forma idónea, asegurando que la producción de café alcance a cubrir las necesidades básicas y compromisos en que incurren los caficultores.

Actualmente en las fincas cafeteras el control realizado a las actividades básicas de administración del cultivo como los son la contratación, el pago a empleados, la compra de insumos y el cálculo de producción de los lotes se realiza de forma empírica, es decir, los caficultores basan sus actividades en la experiencia de sus padres o en la propia, ocasionando que las fincas hoy se administren de la misma forma en que se realizaba hace 50 años.

El seguimiento y control que los caficultores les realizan a estas actividades agrícolas es manual, utilizan lápiz y papel, provocando que este proceso sea, engorroso, genere pérdida de información y falta de control sobre la producción, los costos y las ganancias de cada lote; además convirtiéndose en un impedimento al momento de realizar comparaciones con datos históricos que ayude a una mejor toma de decisiones sobre los procedimientos aplicados en la finca, impidiendo la optimización de sus procesos administrativos.

La sistematización de actividades permite que se desarrollen nuevas formas de administrar las fincas, ya que optimiza la obtención y análisis de los datos mejorando la toma de decisiones y retroalimentando la planificación, seguimiento, control y evaluación de los procesos.

Un sistema de información centralizado permite esta sistematización ya que los datos se encuentran almacenados y a disposición de los dueños o administradores en el momento en que ellos lo requieran, dando la posibilidad de realizar evaluaciones respecto a la situación actual de los lotes ayudando a predecir datos de cosechas venideras a través de información que se encuentra actualizada, permitiendo la contratación de personal más idóneo y productivo para cada una de las actividades que requiere el cultivo, mejorando los procesos administrativos de la finca.

3. OBJETIVO GENERAL

Desarrollar un prototipo de un sistema de información orientado a la web para la finca cafetera AltoBonito ubicada en el Departamento de Caldas, Municipio Villamaria, Vereda Miraflores

4. OBJETIVOS ESPECÍFICOS

- Levantar y analizar los requerimientos.
- Analizar, diseñar, implementar y probar el módulo de acceso y registro de usuarios
- Analizar, diseñar, implementar y probar el módulo ingreso de trabajadores y producción.
- Analizar, diseñar, implementar y probar el módulo de administración de lotes.
- Analizar, diseñar, implementar y probar el módulo de análisis de rentabilidad.
- Elaborar de manual técnico del prototipo.

5. ALCANCE

En este documento se muestra el análisis y diseño de un sistema de implementación de un sistema de información para la finca cafetera AltoBonito ubicada en el departamento de Caldas, Municipio de Villamaria. Las técnicas de ingeniería de software que se utilizaron son establecimiento de requerimientos, vistas 4+1, implementación mediante una metodología ágil de desarrollo, creación de planes de prueba y diseño de manual técnico.

La implementación del prototipo abarca las siguientes funcionalidades las cuales se describe por módulos:

- Módulo de acceso y registro de usuarios.
- Módulo de administración de lotes
- Módulo de ingreso de trabajadores y producción
- Módulo de análisis de rentabilidad.

6. MARCO TEÓRICO

6.1 INGENIERÍA DE SOFTWARE

Actualmente la sociedad se encuentra en proceso de digitalizar toda la información, las industrias implementan cada día software de control en cada uno de sus procesos, esto con el fin de agilizar tareas y sobre todo para el seguimiento y análisis de resultados, bajo este marco es necesario que aparezca la ingeniería del software como método para producir software costeable y asequible por los millones de usuarios a nivel global de las nuevas tecnologías

6.2 ¿QUÉ ES LA INGENIERÍA DE SOFTWARE?

El IEEE Computer Society define la ingeniería de software como:

"Aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, la aplicación de la ingeniería del software"²

"Es una disciplina de la ingeniería que permite comprender todos los aspectos de la producción de software, desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de este después de que se utiliza"³

La ingeniería de software comprende además de los procesos técnicos del desarrollo del software, actividades tales como gestión de proyectos de software y el desarrollo de herramientas, métodos y teorías de apoyo a la producción de software.

La ingeniería de software se encuentra organizada en 15 áreas del conocimiento, según el proyecto liderado por la IEEE Computer Society, proyecto denominado swebok, cada una de estas áreas está descrita por una visión general de su ámbito y como se relaciona con las demás áreas del conocimiento, consta de sub-áreas y sub-temas que ofrecen un marco amplio de conocimiento de los tópicos enmarcados dentro de cada una para describir a gran nivel de detalle cómo funciona cada parte de la ingeniería de software⁴.

Las 15 áreas de conocimiento son:

- Requisitos de software.
- Diseño de software.
- Construcción de software.
- Pruebas de software.
- Mantenimiento de software.

² IEEE Standard glossary of software engineering terminology, IEEE std 610.12-1990.1990

³ Sommerville Ian, Ingeniería del software 7ma. Ed.-, Pearson education S.A, séptima edición, p. 6

⁴ <http://www.computer.org/web/swebok>

- Gestión de la configuración.
- Gestión de la ingeniería de software.
- Proceso de la ingeniería de software.
- Herramientas y métodos de la Ingeniería de Software.
- Calidad del software.
- Práctica profesional de la Ingeniería de Software.
- Economía de la Ingeniería de Software.
- Fundamentos de Computación.
- Fundamentos matemáticos.
- Fundamento de la Ingeniería.

6.3 ¿QUE ES EL PROCESO SE SOFTWARE?

Es el conjunto de actividades y resultados asociados que producen el producto de software, principalmente se encuentran 4 actividades para describir este proceso⁵:

1. Especificaciones de software, el cliente y el ingeniero de software definen que software se va a producir, las condiciones y restricciones de su operación.
2. Desarrollo de software, se realiza el diseño y codificación del programa.
3. Validación de software, se prueba el software para asegurarse de que el producto es lo que el cliente quiere.
4. Evaluación del software, modificaciones al software para adaptarlo a los cambios requeridos por parte del mercado y de los clientes.

Existen diferentes tipos de sistemas de información, dependiendo de la naturaleza y objetivo de estos se requieren igualmente distintos procesos para su desarrollo. Por ejemplo algunos sistemas deben ser totalmente especificados antes de empezar su desarrollo como lo son los de “tiempo real”, sin embargo se encuentran otros que pueden ser desarrollados de manera modular, permitiendo probar lo realizado mediante la implantación progresiva del sistema, dando pie a mejoras continuas, por lo tanto es necesario elegir un proceso de desarrollo de software adecuado a las características del producto que se va a desarrollar, puesto que elegir un proceso que no cumple con los lineamientos del producto puede retrasar su consecución y comprometer la calidad del trabajo final.

⁵ Sommerville Ian, Ingeniería del software 7ma. Ed.-, Pearson education S.A, séptima edición pag 7

6.4 METODOLOGÍAS AGILES

Las metodologías ágiles o desarrollo ágil de software, son un concepto que se refiere al conjunto de métodos y estándares utilizados para un desarrollo incremental o iterativo de un producto de software, donde los requerimientos van evolucionando a través de un conjunto de prácticas organizadas y estructuras en grupos de colaboración multidisciplinarios que realizan el producto mediante iteraciones, las cuales tienen un ciclo de vida que incluye etapas conocidas como planificación, análisis, diseño, codificación, pruebas y documentación.

Existen numerosas metodologías para el desarrollo de software que se describen como ágiles, estas utilizan diferentes métodos y principios en su camino por satisfacer las necesidades del sistema que buscan implementar. Algunas de las más importantes y populares son:

- DYNAMIC SYSTEMS DEVELOPMENT METHOD (DSDM)
- CRYSTAL METHODOLOGIES
- ADAPTATIVE SOFTWARE DEVELOPMENT (ASD)
- EXTREME PROGRAMMING (XP)
- SCRUM

Para la realización de este proyecto de grado se utilizará la metodología SCRUM que permite que el proceso de construcción de software se vuelva un proceso rápido y enfocado a los objetivos que se tienen establecidos.

6.4.1 METODOLOGÍA SCRUM

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo y obtener el mejor resultado posible en un proyecto.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Ésta metodología ágil es muy usada para proyectos que presenten entornos complejos, que requieran resultados tempranos y en el cual sus requisitos sean cambiantes o poco definidos⁶.

Los beneficios de aplicar esta metodología en un proyecto se pueden ver reflejados en aspectos como⁷:

- **Entrega regulares de resultados:** Es decir, el cliente puede empezar a ver y usar los resultados antes de que el proyecto se encuentre finalizado.

⁶ <http://www.proyectosagiles.org/que-es-scrum>

⁷ <http://www.proyectosagiles.org/beneficios-de-scrum>

- **Productividad y calidad:** De manera regular el equipo va mejorando la calidad de trabajo.
- **Flexibilidad y adaptación:** de manera regular el cliente redirige el proyecto en función a las nuevas prioridades.
- **Retorno de inversión (ROI):** De manera regular el cliente maximiza el ROI del proyecto.
- **Mitigación de los riesgos:** Desde el inicio del proyecto el equipo tiene que gestionar los problemas que puedan aparecer en una entrega posterior.

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos, normalmente son iteraciones de un mes y hasta de dos semanas, si así se necesita. Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

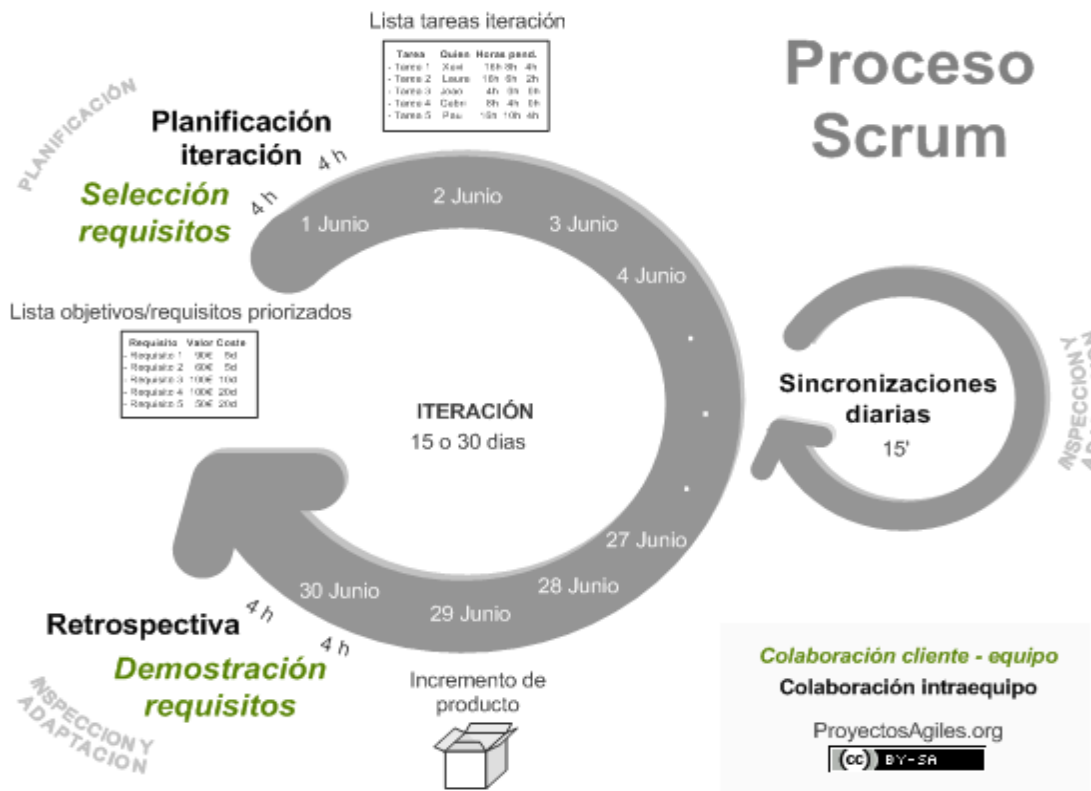


Ilustración 1 Proceso Scrum (véase en www.proyectosagiles.org)

El proceso parte de la lista de requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente prioriza los objetivos balanceando el valor que le aportan respecto a su coste y quedan repartidos en iteraciones y entregas. De manera regular el cliente puede maximizar la utilidad de lo que se desarrolla y el retorno de inversión mediante la re planificación de objetivos del producto, que se realiza durante la iteración con vista a las siguientes iteraciones. Cada iteración dentro de la metodología SCRUM se define como un sprint.

SCRUM permite cuatro eventos formales, contenidos dentro de sprint⁸:

1. Reunión de Planificación de Sprint (Sprint Planning Meeting)
2. Scrum Diario (Daily Scrum)
3. Revisión del Sprint (Sprint Review)
4. Retrospectiva del Sprint (Sprint Retrospective)

El equipo SCRUM:

- Product owner o dueño del producto
- Scrum Master
- Equipo de desarrollo

Artefactos de SCRUM

- Lista del producto (product backlog): La lista del producto es una lista ordenada de todo lo que podría ser necesario en el producto, y es la única fuente de requisitos para cualquier cambio a realizar en el producto.
La lista de producto enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que constituyen cambios a ser hechos sobre el producto para entregas futuras. Los elementos de la Lista de Producto tienen como atributos la descripción, la ordenación, la estimación y el valor.

6.5 LEVANTAMIENTO Y ANÁLISIS DE REQUERIMIENTOS

Los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas, estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema, el proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se denomina ingeniería de requerimientos RE⁹.

Requerimientos del usuario: Declaraciones en lenguaje normal y diagramas, de servicios que se espera que el sistema brinde, al igual que las restricciones que se deben tener.

Requerimientos funcionales: Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema debe hacer.

⁸ Schwaber K. Sutherland J, "La guía de scrum" disponible en:

<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf> p. 5

⁹ Sommerville Ian, Ingeniería del software 7ma. Ed.-, Pearson education S.A, séptima edición, p. 108.

Requerimientos no funcionales: Son restricciones de los servicios o funciones ofrecidos por el sistema, incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares¹⁰.

El proceso de obtención y análisis de requerimientos puede ser clasificado en las siguientes fases:

Descubrimiento: Proceso de interactuar con los interesados del sistema y recopilar sus requerimientos.

Clasificación: Organización en grupos coherentes, por prioridades, funcionales y no funcionales, entre otros.

Documentación: Se debe realizar una documentación clara de los requerimientos recabados, ya que estos son la base para empezar el proceso de diseño del sistema.

6.5.1 HISTORIAS DE USUARIO

Es una técnica utilizada en el desarrollo de software, consiste en la redacción de un requerimiento en unas pocas frases, adicional a esto se utiliza el lenguaje común del usuario.

Las historias de usuario son ideales para desarrollos con metodologías ágiles, definen las especificaciones del proyecto, posteriormente y debido a la prioridad asignada a cada una, se acomodan dentro de las diferentes iteraciones y se realizan los estimados en tiempos para la realización de cada historia, este proceso lo realizan los desarrolladores.

Características de una historia de usuario¹¹:

Independiente: Pueden existir historias con tengan dependencias con otras, es necesario separar de manera que resulten lo más independiente posible, ya que esta independencia es clave a la hora de la etapa del diseño.

Negociable: Las pruebas de validación son la manera en que los clientes pueden verificar el alcance de la historia de usuario.

Valoradas por clientes o usuarios: Las historias de usuarios son importantes para clientes y usuarios, deben reflejar lo que ellos esperan, no lo que el desarrollador quiera hacer.

Estimables: Cuando se estiman los tiempos para cada historia de usuario es más fácil definir cuanto tiempo se tardará un proyecto.

Pequeñas: Deben ser pequeñas por la naturaleza iterativa de los proyectos donde se utilizan.

¹⁰ Sommerville Ian, Ingeniería del software 7ma. Ed.-, Pearson education S.A, séptima edición, p. 110

¹¹ Mike Cohn, 2004"User Stories Applied", , Addison Wesley

Verificables: Debe ser posible la verificación de los logros obtenidos en las historias de usuario en cada entregable de los proyectos.

Las historias de usuario responden a tres preguntas:

¿Quién se beneficia? ¿Qué se quiere? ¿Cuál es el beneficio?

6.6 DISEÑO DE SOFTWARE

Una vez se tienen los requisitos del sistema que se construirá los ingenieros de software pasan a la fase de diseño, en esta y gracias a técnicas como el lenguaje unificado de modelado UML, se realiza el proceso de crear diagramas que especificaran desde como interactuara el sistema con los usuarios, hasta los componentes físicos que tendrá el mismo.

Esta actividad juega un rol vital en el desarrollo de software, debido a que es realmente en la etapa de diseño que un ingeniero de software descubre si es posible realizar un requerimiento tal y como lo específico el cliente.

6.6.1 LENGUAJE UNIFICADO DE MODELADO UML

UML es un conjunto de notaciones graficas que ayudan en la descripción de sistemas de software. El UML nació de la unificación de varios gráficos de modelado orientados a objetos UML ofrece 9 tipos de diagramas ¹²:

- **Diagramas de casos de uso:** modelar procesos.
- **Diagramas de secuencia:** modelar el paso de mensajes entre objetos.
- **Diagramas de colaboración:** modelar interacciones entre objetos.
- **Diagrama de estados:** modelar comportamiento de los objetos en el sistema.
- **Diagramas de actividades:** modelar el comportamiento de los casos de uso, objetos u operaciones.
- **Diagrama de clases:** modelar estructura estática de las clases del sistema.
- **Diagrama de objetos:** modelar la estructura estática de los objetos del sistema.
- **Diagrama de componentes:** modelar componentes.
- **Diagrama de implementación:** modelar la distribución del sistema.

¹² Fowler, M. (2004) *UML distilled: a brief guide to the standard object modeling language*. Addison-Wesley Professional.

6.6.2 VISTAS 4+1

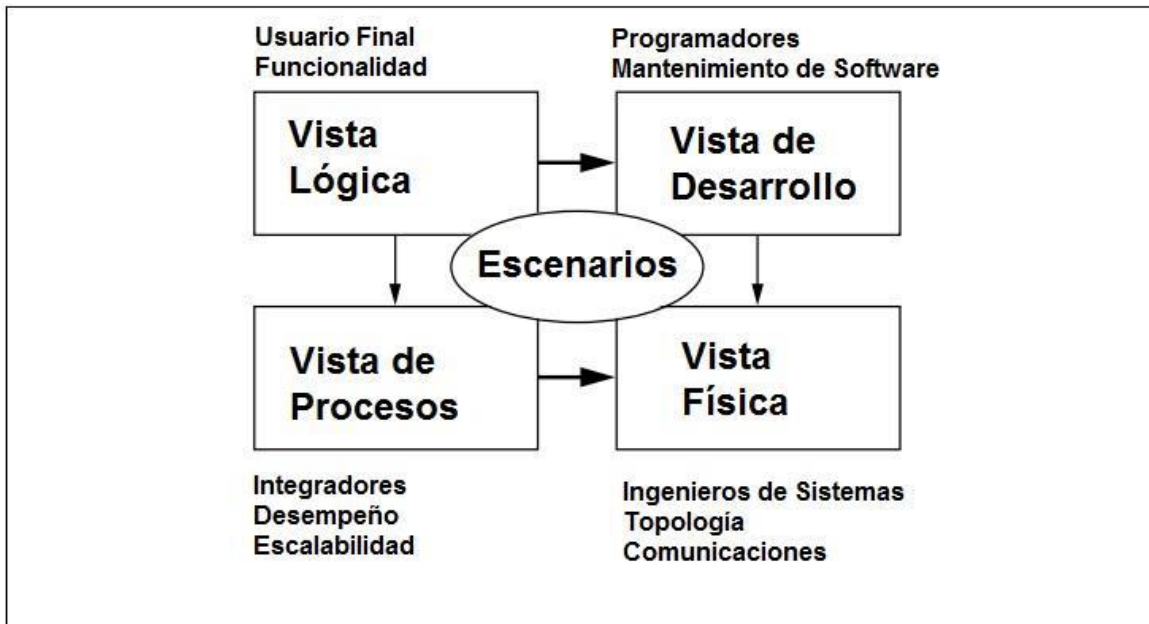


Ilustración 2 Vistas 4+1

La arquitectura de software trata con la abstracción, con descomposición y composición, con estilo y estética, para describir una arquitectura de software se utiliza un modelo compuesto por múltiples vistas o perspectivas. En orden de abordar eventualmente arquitecturas grandes y desafiantes, el modelo propuesto se compone de 5 vistas principales, vistas en la ilustración anterior¹³.

- **La vista Lógica:** es el modelo del objeto del diseño.
- **La vista de procesos:** captura los aspectos de concurrencia y sincronización del diseño.
- **La vista física:** describe el mapeo del software en el hardware, reflejando su aspecto distribuido.
- **La vista de desarrollo:** describe la organización estática del software en su entorno de desarrollo.
- **Escenarios:** esta muestra los elementos de las 4 vistas trabajando juntos sin problemas con el uso de un pequeño conjunto de importantes escenarios -instancias o generalmente casos de uso.

Las vistas 4 + 1 ofrecen una gran ayuda en el desarrollo de un producto de software pues nos llevan paso a paso en la construcción de todos los componentes que tendrá el sistema que se desea crear, empezando con los casos de uso que nos muestran como interactuaran los usuarios del sistema con el mismo, en diferentes situaciones, todo esto dependiente del tipo de usuario y privilegios dentro del sistema con que este cuente.

¹³ Krutchen Phillipe, Rational Software Corp (1995). Architectural Blueprints- The "4+1" View Model of Software Architecture

Con la descripción de la secuencia de actividades que se realiza en cada caso de uso se ve la relación entre el usuario y las respuestas que dará el sistema de manera más detallada, es decir este tipo de diagramas nos permiten detallar como debe responder nuestro sistema a cada petición de un usuario.

Después de tener la interacción de los usuarios con el sistema este modelo de vistas 4 + 1 nos da la posibilidad de detallar los componentes que tendrá el sistema físicos y lógicos: como estará este sistema descrito a nivel de hardware y software con componentes como framework, lenguajes de programación, y por otro lado interfaces de usuario, bases de datos, servidores y otros.

6.7 ARQUITECTURA DE SOFTWARE

La Arquitectura de Software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones¹⁴.

6.7.1 MODELO VISTA CONTROLADOR

El patrón Modelo-Vista-Controlador (MVC) surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. El patrón MVC es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo. A partir del uso de frameworks basados en el patrón MVC se puede lograr una mejor organización del trabajo y mayor especialización de los desarrolladores y diseñadores¹⁵.

Utilizar el patrón modelo vista-controlador para los desarrollos de productos de software asegura una integridad de los datos, debido a que con este modelo se separan las vistas de los usuarios de los controladores de las mismas, es decir primero se asegura que cada usuario solo tendrá acceso dentro del sistema a las partes a las que efectivamente pueda acceder con este modelo se puede controlar esto, además cada vista tiene su controlador y sus propias acciones sobre el sistema.

¹⁴ Paul Clements. (1996) "A Survey of Architecture Description Languages". Proceedings of the International Workshop on Software Specification and Design, Alemania

¹⁵ González, Y. D., & Romero, Y. F (2012). Patrón Modelo-Vista-Controlador. Revista Telemática, 11(1), 47-

Los patrones a la hora de desarrollar software son una tendencia que se debe seguir ya que ayudan en la integridad del software, además contribuyen a que la funcionalidad requerida del sistema se cumple a cabalidad.

Un factor muy importante en cualquier desarrollo de un sistema de software, es la seguridad de los datos y quien puede acceder a ellos el modelo vista controlador ayuda en esta tarea de salvaguardar la información pues a través de su modelo de vista y controlador los usuarios no tienen acceso a toda la información que pueda tener dicha base de datos y esto hace el sistema menos vulnerable a ataques.

6.8 CODIFICACIÓN

La construcción del software en el paso a seguir, una vez se tiene modelado el sistema a construir y definido el tipo de arquitectura, se continúa con la actividad de plasmar todo el análisis y diseño en código, tarea también conocida como implementación.

Pruebas

La tarea de testing es crucial debido a que mide la efectividad de los desarrollos realizados, ayudando a detectar defectos y fallas.

- **Defectos:** errores internos, los encuentran los desarrolladores
- **Fallas:** vista externa y es detectado por usuarios

Probar un software consiste en realizar un conjunto de test sobre los requerimientos funcionales y no funcionales que debe cumplir de acuerdo con lo descrito en el nivel de análisis, de esta manera verificar que fallos y defectos se encuentran en este para realizar posibles mejoras antes de la entrega final del producto.

Características de testing:

- Se debe tener una definición clara de los resultados esperados para la prueba
- Los códigos deben ser idealmente probados por personas ajenas a su desarrollo
- Se deben probar entradas válidas y no válidas para el software

6.9 ESTÁNDARES WEB

Un estándar es un conjunto de reglas normalizadas que describen los requisitos que deben ser cumplidos por un producto, proceso o servicio, con el objetivo de establecer un mecanismo base para permitir que distintos elementos hardware o software que lo utilicen, sean compatibles entre sí. Por lo tanto, si se hace una analogía, los estándares web son un conjunto de reglas normalizadas con el propósito de estandarizar el desarrollo web y posterior ofrecimiento de productos, procesos o servicios web con el fin de lograr que este sea fácil de usar y que todo el mundo pueda confiar. Con estas tecnologías abiertas y de uso libre se pretende evitar la fragmentación de la Web y mejorar las infraestructuras para que se pueda evolucionar hacia una Web con la información mejor organizada¹⁶.

6.9.1 HTML (HYPER TEXT MARKUP LANGUAGE)

Lenguaje compuesto de una serie de etiquetas o marcas que permiten definir el contenido y apariencia de las páginas web. Aunque se basa en el estándar SGML, no se puede considerar que sea un subconjunto de él, existen cientos de etiquetas con diferentes atributos. W3C¹⁷ se encarga de su estandarización¹⁸.

6.9.2 XML (EXTENSIBLE MARKUP LANGUAGE)

Metalenguaje de etiquetado basado en SGML. diseñado específicamente para la WWW por W3C. Permite que un usuario diseñe sus propias etiquetas, con sus atributos y las reglas de construcción de documentos (sintaxis)¹⁹.

6.9.3 JAVASCRIPT

Es el lenguaje interpretado orientado a objetos desarrollado por Netscape que se utiliza en millones de páginas web y aplicaciones de servidor en todo el mundo²⁰.

¹⁶ Guía breve sobre estándares web- disponible en
<<http://www.w3c.es/Divulgacion/GuiasBreves/Estandares>>

¹⁷ W3C HTML disponible en: <http://www.w3.org/html/>

¹⁸ MORA, Sergio Lujan,2002, Programación de aplicaciones web: historia, principios básicos y clientes web, p. 23.

¹⁹ MORA, Sergio Lujan,2002, Programación de aplicaciones web: historia, principios básicos y clientes web, p. 28.

²⁰ Que es JavaScript disponible en:
https://developer.mozilla.org/es/docs/Web/JavaScript/Acerca_de_JavaScript

Este lenguaje permite crear aplicaciones que se ejecutan a través de internet. Basada en el paradigma cliente servidor. La parte de cliente se ejecuta en el navegador, mientras la parte servidor se ejecuta en un servidor de las compañías²¹.

6.9.4 CSS (CASCADING STYLE SHEETS)

Hojas de Estilo en Cascada, es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento²².

6.9.5 PHP (PHP: HYPERTEXT PREPROCESSOR)

Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo²³.

²¹ ¿Qué es Javascript? Disponible en: <http://www.desarrolloweb.com/javascript/#quees>

²² Guías breves de CSS disponible en: <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>

²³ Que es php disponible en: <http://php.net/manual/es/intro-what-is.php>

6.9.6 FRAMEWORK

Una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta²⁴.

6.9.7 CODEIGNITER

CodeIgniter es un Framework para el Desarrollo de Aplicaciones, una herramienta para la gente que crea webs usando PHP. Su meta es permitir desarrollar proyectos mucho más rápido que si se hiciera escribiendo el código desde cero, proporcionando una gran variedad de librerías para las tareas más corrientes, así como una interfaz simple y una estructura lógica para acceder. Codeigniter permite minimizar la cantidad de código necesaria para una tarea determinada²⁵.

Una de las grandes ventajas de este framework es su facilidad de configuración

6.9.8 MY SQL

MySQL es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento.

Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo²⁶.

²⁴ ¿Qué es un framework web? Disponible en:

http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

²⁵ Guía del usuario de Codeigniter versión 2.1.4 disponible en: http://escodeigniter.com/guia_usuario/

²⁶ Casillas Santillán L, Gibert Ginestà M, Pérez Mora O, bases de datos en mysql, disponible en: http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02151.pdf

7. DEFINICIÓN DE LOS REQUERIMIENTOS DEL SISTEMA

Los requerimientos del sistema se construyeron a partir de la técnica de historias de usuario (HU), como lo indica el marco de trabajo SCRUM estos se dividen en iteraciones llamadas SPRINT, cada una de las HU pertenecen a uno de los sprint.

Dentro de la metodología de ingeniería utilizada para el desarrollo del proyecto se dividieron las HU en tres sprint, el documento generado para esto se conoce como product backlog uno de los artefactos de SCRUM. (ver ANEXO 1 product backlog).

7.1 HISTORIAS DE USUARIO REQUERIMIENTOS NO FUNCIONALES

NOTA: estos requerimientos no funcionales se tienen en cuenta durante todos los sprints por lo tanto aparecen referenciados de esta manera y no incluidos dentro de cada sprint.

Tabla 1 Historia de Usuario 1

Historia de Usuario	
Número: 1	Nombre: Navegadores
Prioridad en Negocio:	Iteración Asignada: 1
Descripción: Debe ser un software de arquitectura web en donde los usuarios podrán acceder mediante Internet por los navegadores Internet Explorer 10 o superior, Firefox 23 o superior, Google Chrome 28.0 o superior, además se debe ajustar a diferentes resoluciones de pantalla.	
Observaciones:	

Tabla 2 Historia de Usuario 2

Historia de Usuario	
Número: 2	Nombre: Interfaz de usuario
Prioridad en Negocio:	Iteración Asignada: 1
Descripción: Cumplir con requisitos mínimos de usabilidad de la W3C	
Observaciones:	

Tabla 3 Historia de Usuario 3

Historia de Usuario

Número: 3	Nombre: Desempeño
Prioridad en Negocio:	Iteración Asignada: 1
Descripción: Cada petición que realice el usuario debe tener una respuesta del sistema en máximo 3 segundos y soportar al menos 1000 usuarios conectados de forma simultánea.	
Observaciones:	

7.2 SPRINT 1

7.2.1 HISTORIAS DE USUARIO

Tabla 4 Historia de Usuario 4

Historia de Usuario	
Número: 4	Nombre: Tipos de Usuario
Prioridad en Negocio:	Iteración Asignada: 1
Descripción: Yo como cliente espero que el sistema permita la creación de tres tipos de usuario: <ul style="list-style-type: none"> • Dueño de finca. • Administrador. • Trabajadores. 	
Observaciones: El perfil dueño de finca es el primer usuario en registrarse además solo puede existir un dueño de finca y es este el que asigna el perfil administrador.	

Tabla 5 Historia de Usuario 5

Historia de Usuario	
Número: 5	Nombre: Registro de Dueño de Finca
Prioridad en Negocio:	Iteración Asignada: 1
Descripción: Yo como cliente espero que el sistema al momento del registro de un usuario dueño de finca solicite la siguiente información: <ul style="list-style-type: none"> • Tipo de identificación (Cédula, Pasaporte, Cédula de extranjería, Tarjeta de identidad). • Número de identificación. • Nombre. • Apellido. • Fecha de nacimiento. • Edad. 	

<ul style="list-style-type: none"> • Genero.
Observaciones: La edad se debe ser calculada automáticamente por el sistema mediante la fecha de nacimiento.

Tabla 6 Historia de Usuario 6

Historia de Usuario	
Número: 6	Nombre: Funciones perfil Dueño de Finca
Prioridad en Negocio:	Iteración Asignada: 1
Descripción: Yo como cliente espero que el sistema permita al perfil dueño de finca las siguientes acciones: <ul style="list-style-type: none"> • Registrarse en el sistema. • Gestionar sus datos básicos. • Crear un usuario administrador. • Eliminar administrador. 	
Observaciones:	

Tabla 7 Historia de Usuario 7

Historia de Usuario	
Número: 7	Nombre: Registro perfil administrador
Prioridad en Negocio:	Iteración Asignada: 1
Descripción: Yo como cliente espero que el sistema al momento del registro del perfil administrador solicite la siguiente información: <ul style="list-style-type: none"> • Tipo de identificación (Cédula, Pasaporte, Cédula de extranjería, Tarjeta de identidad). • Número de identificación. • Nombre. • Apellido. • Fecha de nacimiento. • Edad. • Genero. • Lote. 	
Observaciones: <ul style="list-style-type: none"> • La edad se debe ser calculada automáticamente por el sistema mediante la fecha de nacimiento. • Para completar el campo de lote, el sistema debe mostrar una lista con los lotes 	

asociados a la finca que aún no tengan administrador. En caso de que no haya ninguno lote disponible el sistema debe permitir la creación del usuario y dejar ese campo con un valor de “Pendiente”

Tabla 8 Historia de Usuario 8

Historia de Usuario	
Número: 8	Nombre: Funciones perfil administrador
Prioridad en Negocio:	Iteración Asignada: 1
Descripción: <p>Yo como cliente espero que el sistema permita al usuario administrador las siguientes acciones:</p> <ul style="list-style-type: none"> • Gestionar sus datos básicos. • Crear perfil trabajador. • Gestionar información básica de perfil trabajador. • Eliminar trabajadores. 	
Observaciones:	

Tabla 9 Historia de Usuario 9

Historia de Usuario	
Número: 9	Nombre: Registro perfil trabajador
Prioridad en Negocio:	Iteración Asignada: 1
Descripción: <p>Yo como cliente espero que el sistema al momento del registro del perfil trabajador solicite la siguiente información:</p> <ul style="list-style-type: none"> • Tipo de identificación (Cédula, Pasaporte, Cédula de extranjería, Tarjeta de identidad). • Número de identificación. • Nombre. • Apellido. • Fecha de nacimiento. • Edad. • Genero. 	
Observaciones: <ul style="list-style-type: none"> • Los datos básicos Nombre, apellido y genero son datos obligatorios mientras que los demás se pueden dejar en blanco. • Este usuario no tendrá acceso directo al aplicativo; su registro, modificación y eliminación será responsabilidad del perfil administrador. 	

Tabla 10 Historia de Usuario 10

Historia de Usuario		
Número: 10	Nombre: Autenticación de usuarios	
Prioridad en Negocio:		Iteración Asignada: 1
Descripción: Yo como cliente deseo que para realizar el proceso de autenticación el sistema solicite un usuario y una contraseña.		
Observaciones:		

Historia de Usuario		
Número: 11	Nombre: Primer inicio de sesión	
Prioridad en Negocio:		Iteración Asignada: 1
Descripción: Yo como cliente deseo que cuando un usuario realice su primer inicio de sesión se despliegue un mensaje en el cual le solicite al usuario su cambio de contraseña y un registro de una pregunta y respuesta secreta.		
Observaciones: <ul style="list-style-type: none"> • Para realizar el primer inicio de sesión el usuario debe ingresar en el campo usuario su número de identificación y en el campo Contraseña su número de identificación. • La pregunta y respuesta secreta se guarda con fines de recuperación de contraseña. 		

Tabla 11 Historia de Usuario 11

Tabla 12 Historia de Usuario 12

Historia de Usuario		
Número: 12	Nombre: Recuperar Contraseña	
Prioridad en Negocio:		Iteración Asignada: 1
Descripción: Yo como cliente deseo que el sistema permita modificar la contraseña en caso de ser olvidada.		
Observaciones: Este proceso se realizará respondiendo una pregunta secreta, previamente establecida por el usuario.		

7.2.2 VISTA DE ESCENARIOS (CASOS DE USO)

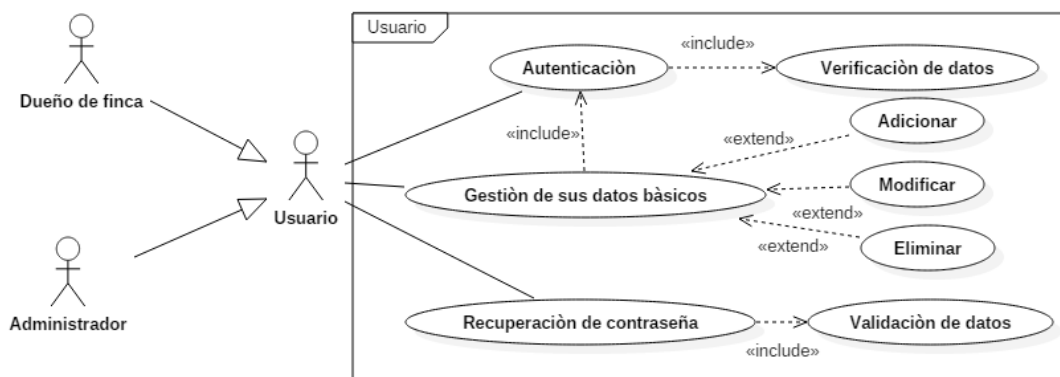


Ilustración 3 Diagrama de Caso de Uso: Usuario Registrado

Tabla 13 Caso de Uso 1

Caso de uso # 1	
Caso de uso	Autenticación
Actores	Dueño de finca, Administrador
Propósito	Ingresa al software
Resumen	Este caso de uso comienza cuando el usuario desea ingresar a una cuenta ya existente en la plataforma.
Prerrequisito	Caso de uso #4
Curso normal de los eventos	
Acción de los actores	Respuesta del sistema
1. El usuario ingresa su usuario y contraseña en la página de inicio.	2. El sistema verifica la información suministrada con la guardada en la base de datos.
	3. El sistema permite el ingreso al usuario y la visualización de su página de inicio.

Curso alterno
Acción 2: Si el usuario ingresa datos erróneos, el sistema debe mostrar un mensaje de “usuario o contraseña incorrecta” y regresar al paso 1.

Tabla 14 Caso de Uso 2

Caso de uso # 2	
Caso de uso	Gestión de sus datos básicos
Actores	Dueño de finca, Administrador
Propósito	Gestionar los datos básicos registrados en el sistema
Resumen	El usuario dueño de finca y administrador puede agregar, editar o eliminar los datos que previamente se encuentran registrados en el sistema, además de realizar el cambio de su contraseña y de la pregunta y respuesta de seguridad.
Prerrequisito	Caso de uso #1
Curso normal de los eventos	
Acción de los actores	Respuesta del sistema
1. El usuario se autentica en el sistema como perfil dueño de finca.	
2. El usuario selecciona la opción “Datos Basicos”	3. El sistema despliega un formulario con los datos básicos registrados en el sistema.
4. El usuario agrega, modifica o elimina la información que desea.	5. El sistema verifica que la información digitada sea correcta para los campos y actualiza la base de datos.
	6. El sistema despliega un mensaje de confirmación.

Curso alterno
Acción 4: El usuario digita datos erróneos, el sistema debe mostrar un mensaje con los errores encontrados y regresar al paso 4.

Tabla 15 Caso de Uso 3

Caso de uso #3	
Caso de uso	Recuperación de contraseña
Actores	Dueño de finca, Administrador
Propósito	Permitir al usuario dueño de finca y administrador recuperar y cambiar su contraseña en caso de que la olvide.
Resumen	Si el usuario llega a olvidar la contraseña de ingreso, tendrá la opción de recuperar la contraseña por medio de una pregunta secreta previamente establecida.
Prerrequisito	Caso de uso #4
Curso normal de los eventos	
Acción de los actores	Respuesta del sistema
1. El usuario selecciona la opción de recuperar contraseña.	2. El sistema despliega un mensaje Solicitando el número de identificación del usuario.
3. El usuario digita su identificación.	4. El sistema valida la existencia del usuario.
	5. El sistema despliega un formulario con una pregunta secreta y un espacio para redactar la respuesta a dicha pregunta.
6. El usuario digita la respuesta	7. El sistema verifica la respuesta.

	8. El sistema despliega un mensaje de confirmación de cambio de contraseña.
Curso alterno	
<p>Acción 3: Si el usuario ingresa un usuario inválido, el sistema debe mostrar un mensaje de “usuario inválido” y regresar al paso 3.</p> <p>Acción 6: El usuario digita una respuesta errores, el sistema debe mostrar un mensaje de “Respuesta incorrecta” y regresar al paso 6</p>	

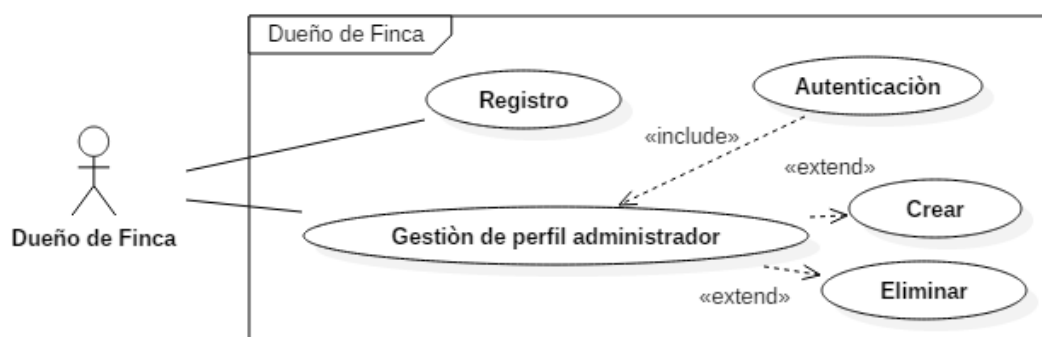


Ilustración 4 Diagrama Caso de Uso: Dueño de Finca

Tabla 16 Caso de Uso 4

Caso de uso # 4	
Caso de uso	Registro
Actores	Dueño de finca
Propósito	Permitir al usuario dueño de finca registrarse en el sistema
Resumen	Este caso de uso comienza cuando el usuario decide crear una cuenta para acceder al sistema, la creación de esta se realiza a través de un formulario.
Prerrequisito	Entrar al sitio web.

Curso normal de los eventos	
Acción de los actores	Respuesta del sistema
1. El usuario ingresa a “Registrar Dueño” en la página de inicio.	2. La aplicación despliega el formulario para el registro.
3. El usuario digita la información solicitada.	4. La aplicación verifica los datos ingresados y los guarda en la base de datos.
	5. La aplicación despliega la información de usuario nuevo.
6. El usuario visualiza un mensaje de confirmación de creación de cuenta.	
Curso alterno	
Acción 4: Si el usuario ingresa datos que no sean consistentes o deja campos incompletos, el sistema muestra un mensaje solicitando los datos y regresa al paso 2.	

Tabla 17 Caso de Uso 5

Caso de uso # 5	
Caso de uso	Gestión de perfil administrador
Actores	Dueño de finca
Propósito	Crear perfil administrador
Resumen	El usuario dueño de finca crea un perfil administrador registrando los datos básicos en el sistema El usuario dueño de finca elimina un perfil administrador
Prerrequisito	Caso de uso #4
Curso normal de los eventos	

Acción de los actores	Respuesta del sistema
1. El usuario se autentica en el sistema como perfil dueño de finca.	
2. El usuario selecciona la opción "Gestión Administrador"	3. El sistema despliega un formulario para el registro del administrador.
4. El usuario digita los datos solicitados en el formulario.	5. El sistema verifica que los datos ingresados sean correctos y los guarda en la base de datos.
	6. El sistema despliega un mensaje de confirmación.
Curso alterno	
Acción 5: Si el usuario ingresa datos erróneos, el sistema debe mostrar un mensaje con los errores encontrados y regresar al paso 4.	

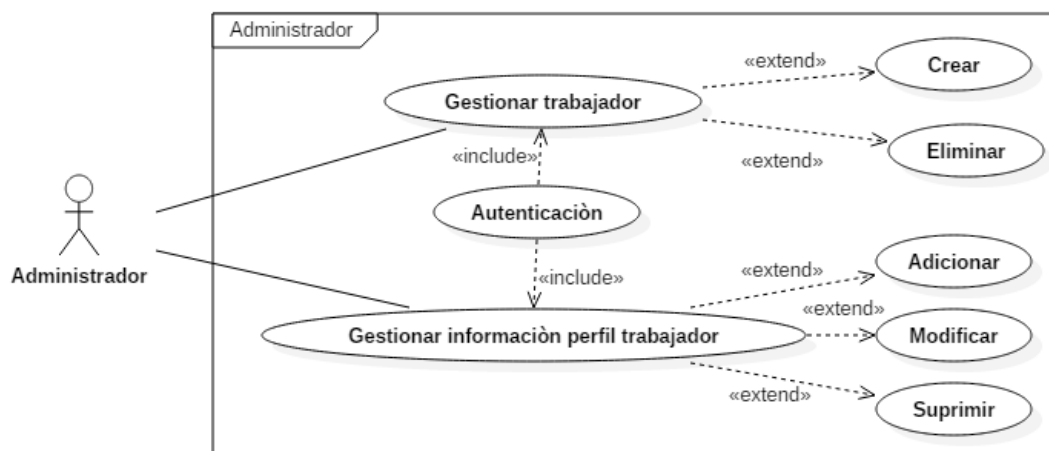


Ilustración 5 Diagrama Caso de Uso: Administrador

Tabla 18 Caso de Uso 6

Caso de uso # 6	
Caso de uso	Gestionar Trabajador
Actores	Administrador
Propósito	Crear y eliminar el perfil trabajador
Resumen	<p>El usuario administrador crea un perfil trabajador registrando los datos básicos en el sistema</p> <p>El usuario administrador elimina un perfil trabajador del sistema</p>
Prerrequisito	Caso de uso #1
Curso normal de los eventos	
Acción de los actores	Respuesta del sistema
1. El usuario se autentica en el sistema como perfil Administrador.	
2. El usuario selecciona la opción "Gestión de Trabajadores"	3. El sistema despliega un formulario con la lista de trabajadores activos para la finca.
4. El usuario selecciona "Nuevo Trabajador" o "Eliminar Trabajador"	5. El sistema despliega un formulario para realizar la creación del trabajador.
6. El usuario digita los datos solicitados.	6. El sistema valida los campos y guarda la información en la base de datos.
	7. El sistema despliega un mensaje de confirmación.
Curso alternativo	
Acción 6: Si el usuario ingresa datos erróneos, el sistema debe mostrar un mensaje con los	

errores encontrados y regresar al paso 6.

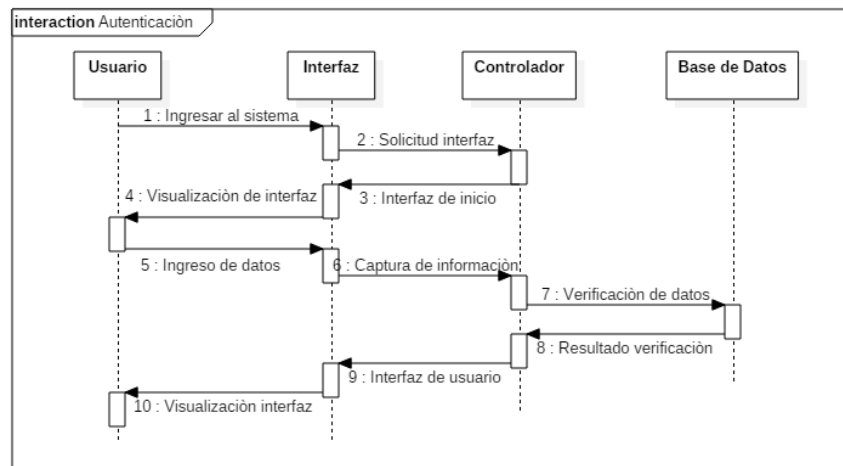
Tabla 19 Caso de Uso 7

Caso de uso # 7	
Caso de uso	Gestionar información del perfil trabajador
Actores	Administrador
Propósito	Gestionar los datos básicos del perfil trabajador.
Resumen	El usuario administrador adiciona, modifica o eliminar los datos básicos del perfil trabajador.
Prerrequisito	Caso de uso #1
Curso normal de los eventos	
Acción de los actores	Respuesta del sistema
1. El usuario se autentica en el sistema como perfil Administrador.	
2. El usuario selecciona la opción "Gestión de Trabajadores"	3. El sistema despliega un formulario con la lista de trabajadores activos para la finca.
4. El usuario selecciona el trabajador al cual desea gestionar	5. El sistema despliega un formulario para realizar la gestión del trabajador.
6. El usuario modifica los datos necesarios.	6. El sistema valida los campos y guarda la información en la base de datos.
	7. El sistema despliega un mensaje de confirmación.
Curso alterno	

Acción 6: Si el usuario ingresa datos erróneos, el sistema debe mostrar un mensaje con los errores encontrados y regresar al paso 6.

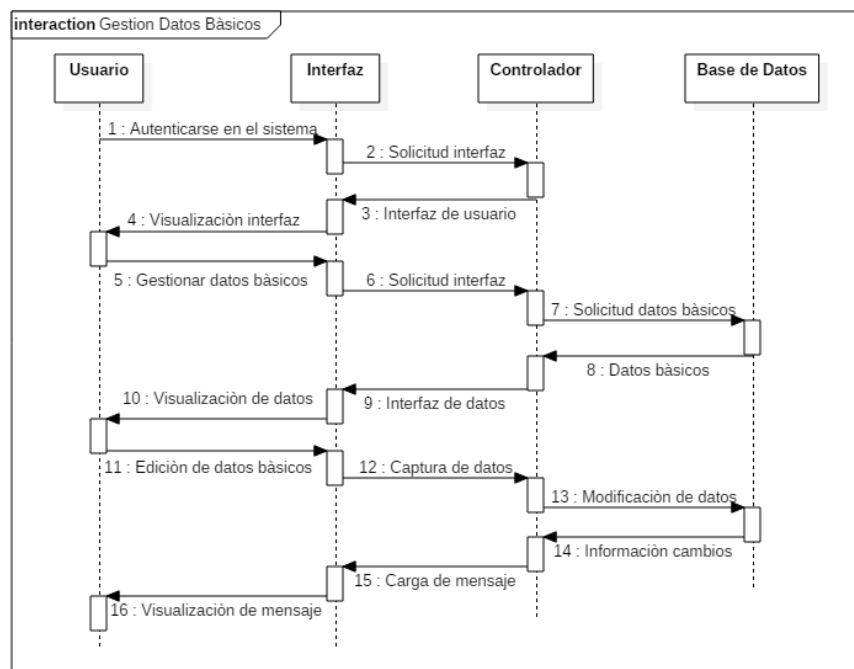
7.2.3 VISTA DE PROCESOS (DIAGRAMAS DE SECUENCIA)

Ilustración 6 Diagrama de Secuencia #1: Autenticación



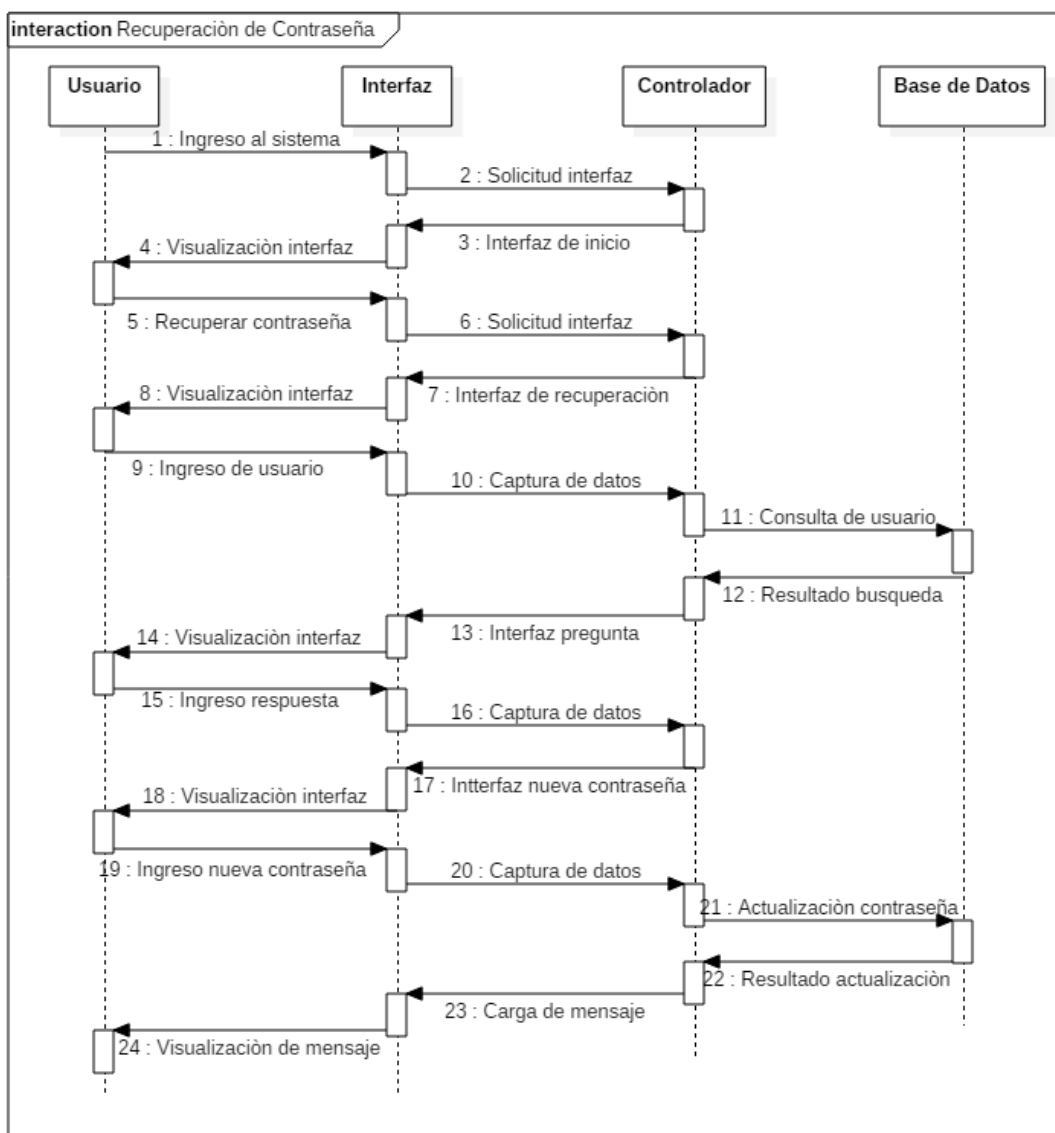
Descripción: Este diagrama muestra la secuencia de acciones a llevar a cabo para realizar la autenticación en el sistema. En caso de que el usuario ingrese datos erróneos el sistema muestra un mensaje de “nombre o contraseña incorrectas” y solicita los datos nuevamente.

Ilustración 7 Ilustración 7 Diagrama de Secuencia #2: Gestión de datos básicos



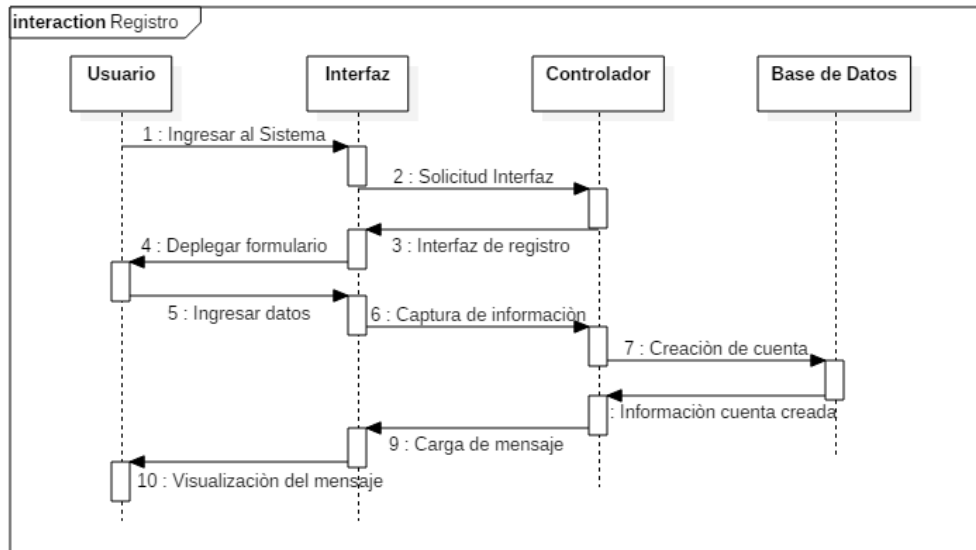
Descripción: Este diagrama muestra la secuencia de acciones a realizar para agregar, modificar o eliminar los datos básicos del usuario previamente guardados en el sistema. En caso de diligenciar de una forma errónea los campos en el formulario el sistema mostrara un mensaje con los errores y solicitara nuevamente los datos.

Ilustración 8 Diagrama de Secuencia #3: Recuperación de contraseña



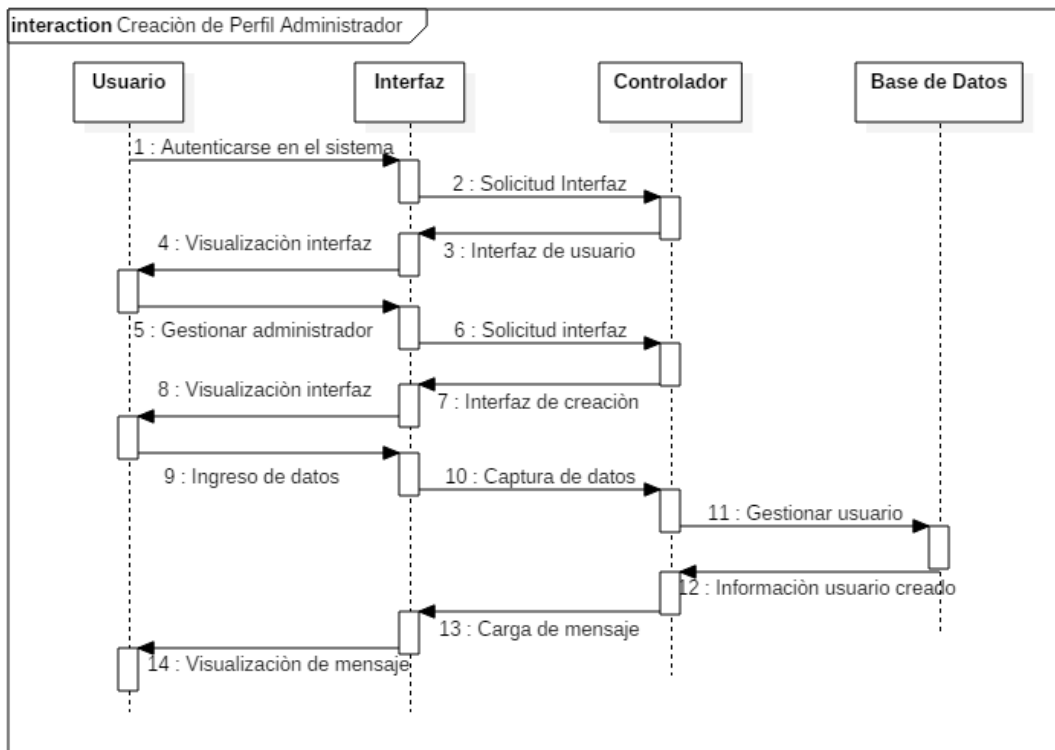
Descripción: Este diagrama muestra el proceso a seguir al momento de realizar el cambio de contraseña. El proceso de recuperación de contraseña se realiza mediante la respuesta a una pregunta secreta que el usuario previamente estableció. En caso de que el usuario digite datos incorrectos en los formularios el sistema mostrara los errores y solicitara nuevamente los datos.

Ilustración 9 Diagrama de Secuencia #4: Registro



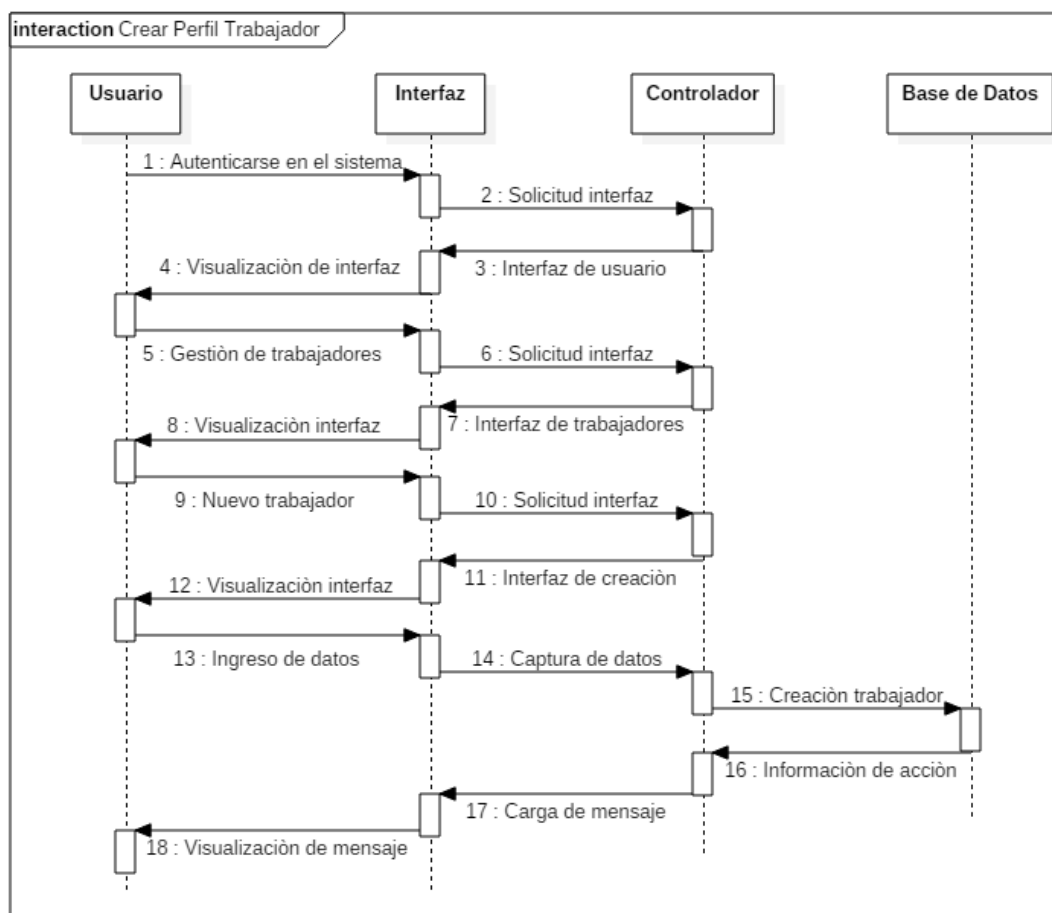
Descripción: Este diagrama muestra la secuencia del caso de uso registro correspondiente al usuario Dueño de Finca. En caso de que el usuario digite mal los campos en los formularios el sistema mostrara los errores y solicitara nuevamente los datos.

Ilustración 10 Diagrama de Secuencia #5: Creación de perfil administrador



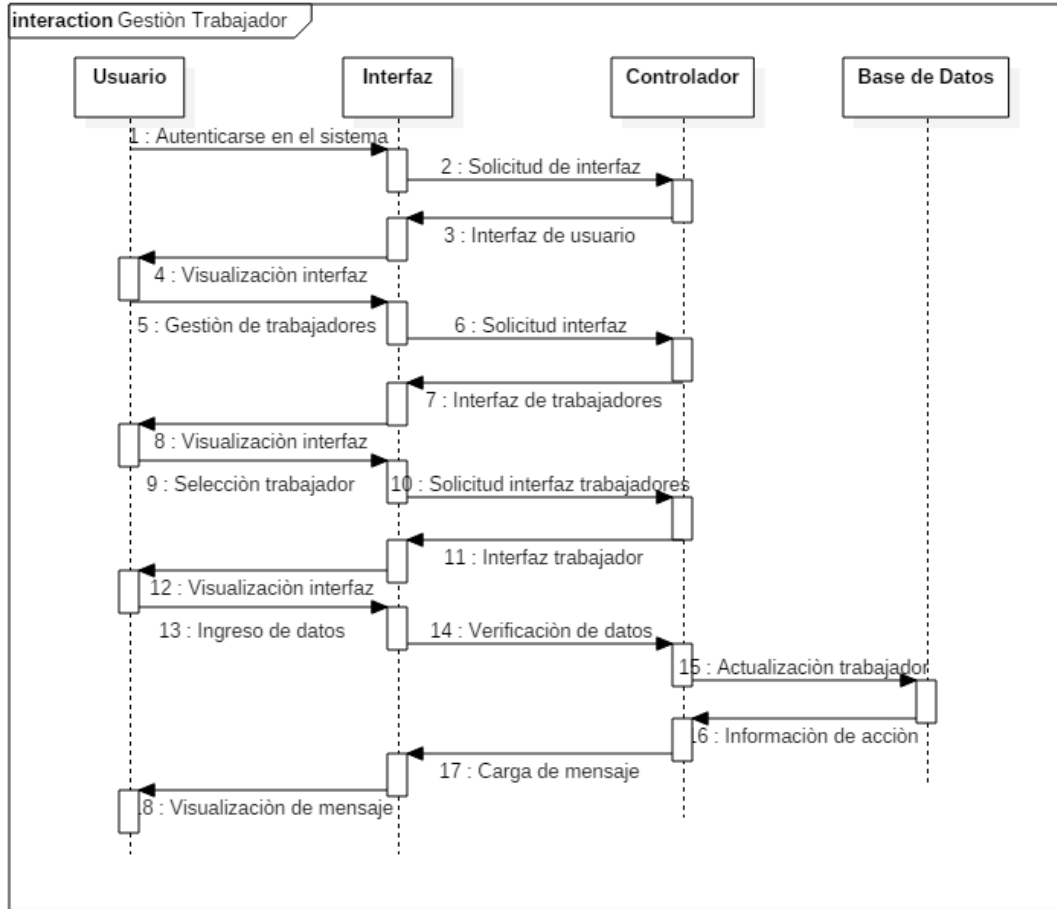
Descripción: Este diagrama muestra el proceso que se debe seguir para que el usuario dueño de finca pueda realizar la gestión del perfil administrador. En caso de que el usuario digite datos erróneos en los formularios el sistema mostrara los errores y solicitara nuevamente los datos.

Ilustración 11 Diagrama de Secuencia #6: Crear perfil trabajador



Descripción: Este diagrama nos muestra la secuencia de pasos a seguir para realizar la adición de un perfil trabajador, el cual se encuentra responsable del perfil Administrador. En caso de que el usuario digite mal los datos en los formularios el sistema mostrara los errores y los solicitara nuevamente.

Ilustración 12 Diagrama de Secuencia #7: Gestionar información del perfil trabajador



Descripción: Este diagrama muestra la secuencia de pasos a seguir para realizar la gestión del perfil Trabajador, el cual se encuentra a cargo del perfil Administrador. En caso de que el usuario cometa errores al diligenciar los campos de los formularios el sistema le mostrara los errores y solicitara nuevamente los datos.

7.3 SPRINT 2

7.3.1 HISTORIAS DE USUARIO

Tabla 20 Historia de Usuario 13

Historia de Usuario	
Número: 13	Nombre: Registro de fincas
Prioridad en Negocio:	Iteración Asignada: 2
Descripción: Yo como cliente espero que el sistema permita al usuario dueño de finca la creación de fincas con la siguiente información: <ul style="list-style-type: none">- Nombre- Tamaño (número de hectáreas cuadradas)- Descripción	
Observaciones: Cuando se elimine la finca no se deben eliminar la lista de administradores y trabajadores asociados a cada uno de los lotes.	

Tabla 21 Historia de Usuario 14

Historia de Usuario	
Número: 14	Nombre: Registro de lotes
Prioridad en Negocio:	Iteración Asignada: 2
Descripción: Yo como cliente espero que el sistema permita al usuario dueño de finca la creación de lotes con la siguiente información: <ul style="list-style-type: none">- Nombre- Tamaño (número de hectáreas cuadradas)- Descripción- Cultivos	
Observaciones: <ul style="list-style-type: none">- Para cada lote se debe asociar una finca, un administrador y cultivos- Al momento de eliminar un lote el tamaño correspondiente al lote debe ser adjudicado a otro(s) lote(s). Cuando se elimine el lote no se deben eliminar el administrador y los trabajadores asociados a este.	

Tabla 22 Historia de Usuario 15

Historia de Usuario	
Número: 15	Nombre: Registro de cultivos
Prioridad en Negocio:	Iteración Asignada: 2
Descripción: <p>Yo como cliente espero que el sistema permita al usuario administrador la creación de cultivos con la siguiente información:</p> <ul style="list-style-type: none"> - Nombre - Número de plantas - Edad - Descripción - Producción 	
Observaciones: <ul style="list-style-type: none"> - Cada cultivo debe tener relacionado labores culturales - Al momento de eliminar un cultivo no se debe eliminar las labores culturales asociadas a esta ni su producción 	

Tabla 23 Historia de Usuario 16

Historia de Usuario	
Número: 16	Nombre: Registro de labores culturales
Prioridad en Negocio:	Iteración Asignada: 2
Descripción: <p>Yo como cliente espero que el sistema permita al usuario administrador la creación de labores culturales con la siguiente información:</p> <ul style="list-style-type: none"> - Nombre - Trabajador responsable - Número de días a laborar - Fecha de finalización - Estado (Pendiente, En progreso, Finalizada) - Herramientas o insumos necesarios - Valor de las herramientas o insumos - Valor realización de labor - Estado de pago (Pendiente, Cancelado) 	
Observaciones:	

7.3.2 VISTA DE ESCENARIOS (CASOS DE USO)

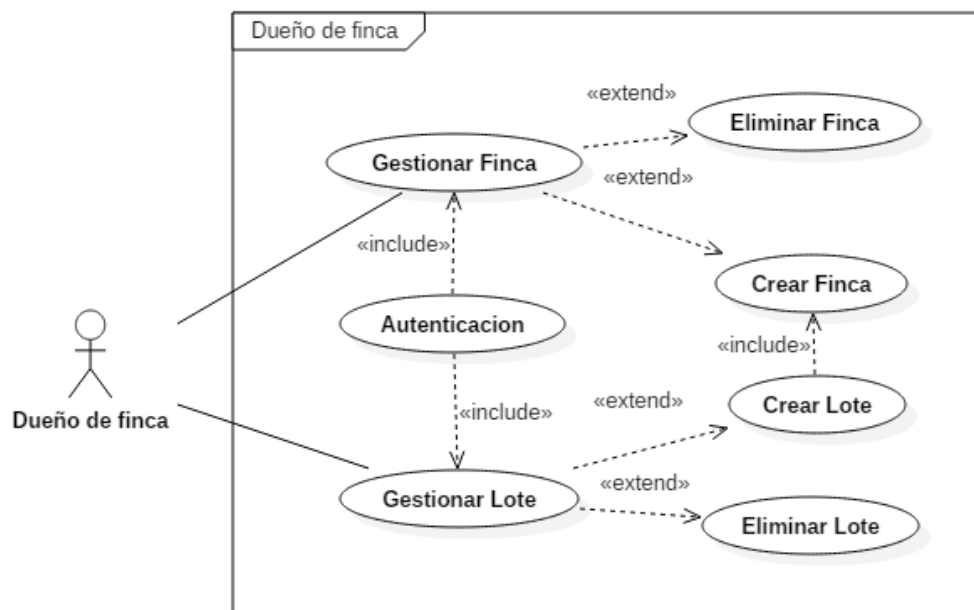


Ilustración 13 Diagrama de Caso de Uso: Dueño de Finca

Tabla 24 Caso de Uso 8

Caso de uso # 8	
Caso de uso	Gestionar Finca
Actores	Dueño de finca
Propósito	Permitir la gestión de fincas
Resumen	Este caso de uso representa la creación y eliminación de fincas
Prerrequisito	Caso de uso #1
Curso normal de los eventos	
Acción de los actores	Respuesta del sistema
1. El usuario se autentica en el sistema como perfil dueño de finca	
2. Ingresar a la pestaña “Gestionar Finca” y	3. El sistema despliega un formulario para

seleccionar “Nueva Finca”	realizar la gestión de la finca.
4. El usuario digita los datos solicitados por el sistema y selecciona “guardar”	5. El sistema verifica los datos y guarda los cambios realizados
Curso alterno	
Acción 2: Si el usuario ingresa datos erróneos, el sistema debe mostrar un mensaje con los datos incorrectos y regresar al paso 4	

Tabla 25 Caso de Uso 9

Caso de uso # 9	
Caso de uso	Gestionar lotes
Actores	Dueño de finca
Propósito	Permitir la gestión de lotes
Resumen	Este caso de uso representa la creación y eliminación de los lotes de una finca
Prerrequisito	Caso de uso #1, #8
Curso normal de los eventos	
Acción de los actores	Respuesta del sistema
1. El usuario se autentica en el sistema como perfil dueño de finca	
2. Ingresar a la pestaña “Administrar Finca” y seleccionar “Agregar Lote”	3. El sistema despliega un formulario para realizar la gestión del lote.
4. El usuario digita los datos solicitados por el sistema y selecciona “guardar”	5. El sistema verifica los datos y guarda los cambios realizados
Curso alterno	

Acción 2: Si el usuario ingresa datos erróneos, el sistema debe mostrar un mensaje con los datos incorrectos y regresar al paso 4

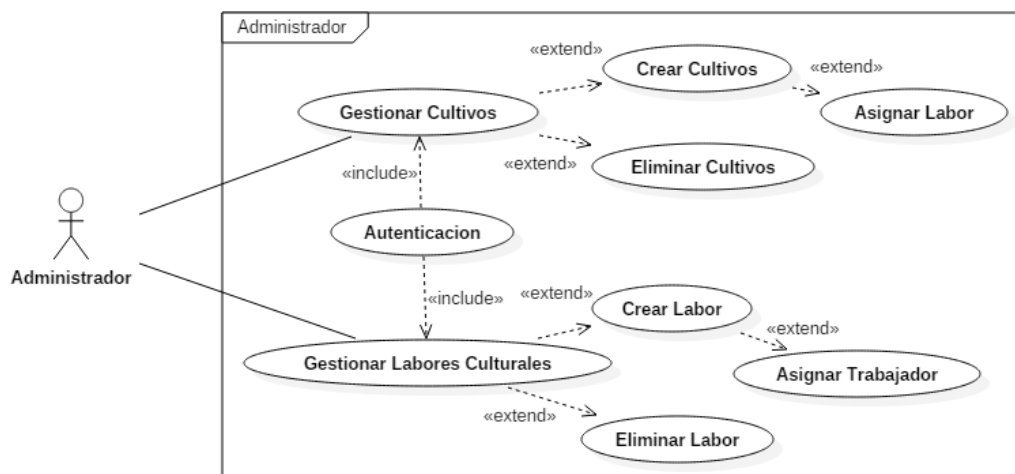


Ilustración 14 Diagrama de Caso de Uso: Administrador

Tabla 26 Caso de Uso 10

Caso de uso # 10	
Caso de uso	Gestionar Cultivos
Actores	Administrador
Propósito	Permitir la gestión de cultivos
Resumen	Este caso de uso representa la creación y eliminación de los cultivos
Prerrequisito	Caso de uso #1, #11
Curso normal de los eventos	
Acción de los actores	Respuesta del sistema
1. El usuario se autentica en el sistema como perfil administrador	
2. Ingresar a la pestaña “Gestionar Cultivos” y	3. El sistema despliega un formulario para

seleccionar “Agregar Cultivo” o “Eliminar Cultivo”	realizar la gestión del lote.
4. El usuario digita los datos solicitados por el sistema y selecciona “aceptar”	5. El sistema verifica los datos y guarda los cambios realizados
Curso alterno	
Acción 2: Si el usuario ingresa datos erróneos, el sistema debe mostrar un mensaje con los datos incorrectos y regresar al paso 4	

Tabla 27 Caso de Uso 11

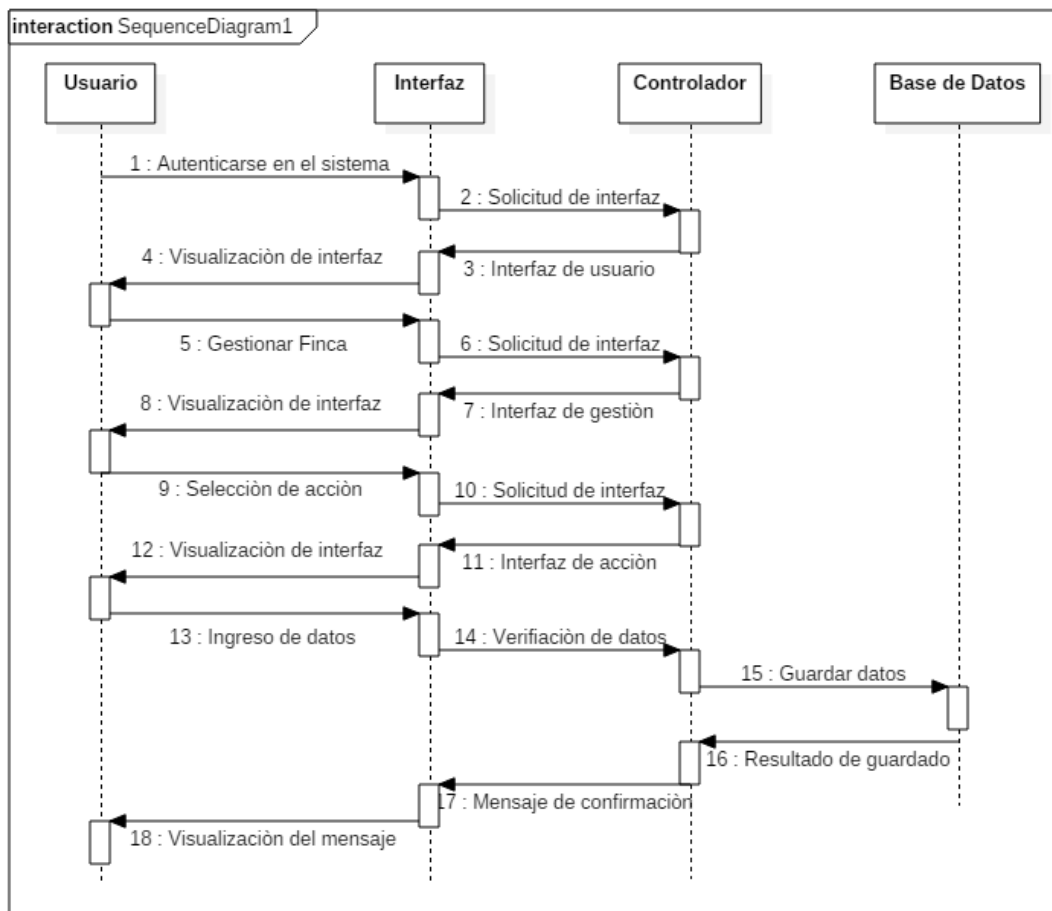
Caso de uso # 11	
Caso de uso	Gestionar Labores Culturales
Actores	Administrador
Propósito	Permitir la gestión de las labores culturales
Resumen	Este caso de uso representa la creación y eliminación de las labores culturales
Prerrequisito	Caso de uso #1, #10
Curso normal de los eventos	
Acción de los actores	Respuesta del sistema
1. El usuario se autentica en el sistema como perfil administrador	
2. Ingresar a la pestaña “Gestionar Labores” y seleccionar “Agregar Labor” o “Eliminar Labor”	3. El sistema despliega un formulario para realizar la gestión del lote.
4. El usuario digita los datos solicitados por el sistema y selecciona “guardar”	5. El sistema verifica los datos y guarda los cambios realizados

Curso alterno

Acción 2: Si el usuario ingresa datos erróneos, el sistema debe mostrar un mensaje con los datos incorrectos y regresar al paso 4

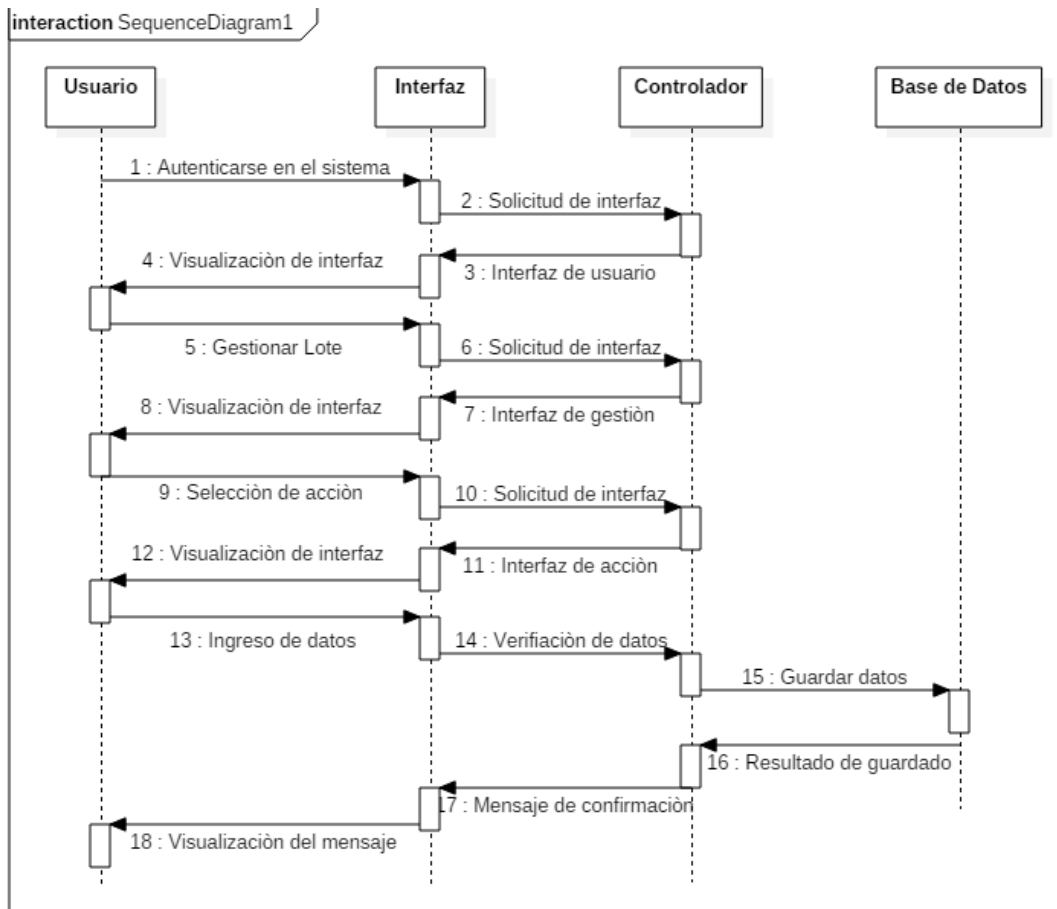
7.3.3 VISTA DE PROCESOS (DIAGRAMAS DE SECUENCIA)

Ilustración 15 Diagrama de Secuencia #8 Gestión de finca



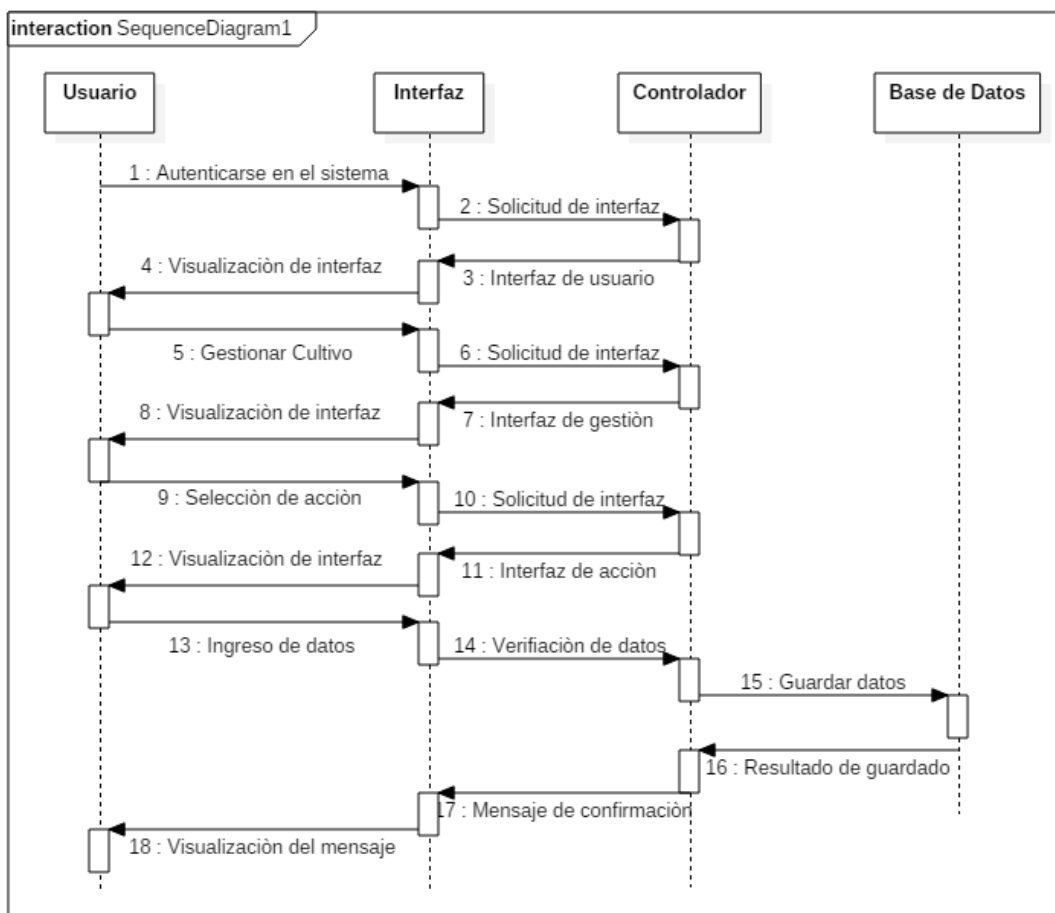
Descripción: Este diagrama nos muestra la secuencia de pasos a seguir para realizar la gestión de una finca, el cual se encuentra responsable el perfil Dueño de Finca. En caso de que el usuario digite mal los datos en los formularios el sistema mostrara los errores y los solicitara nuevamente.

Ilustración 16 Diagrama de Secuencia #9 Gestión de lotes



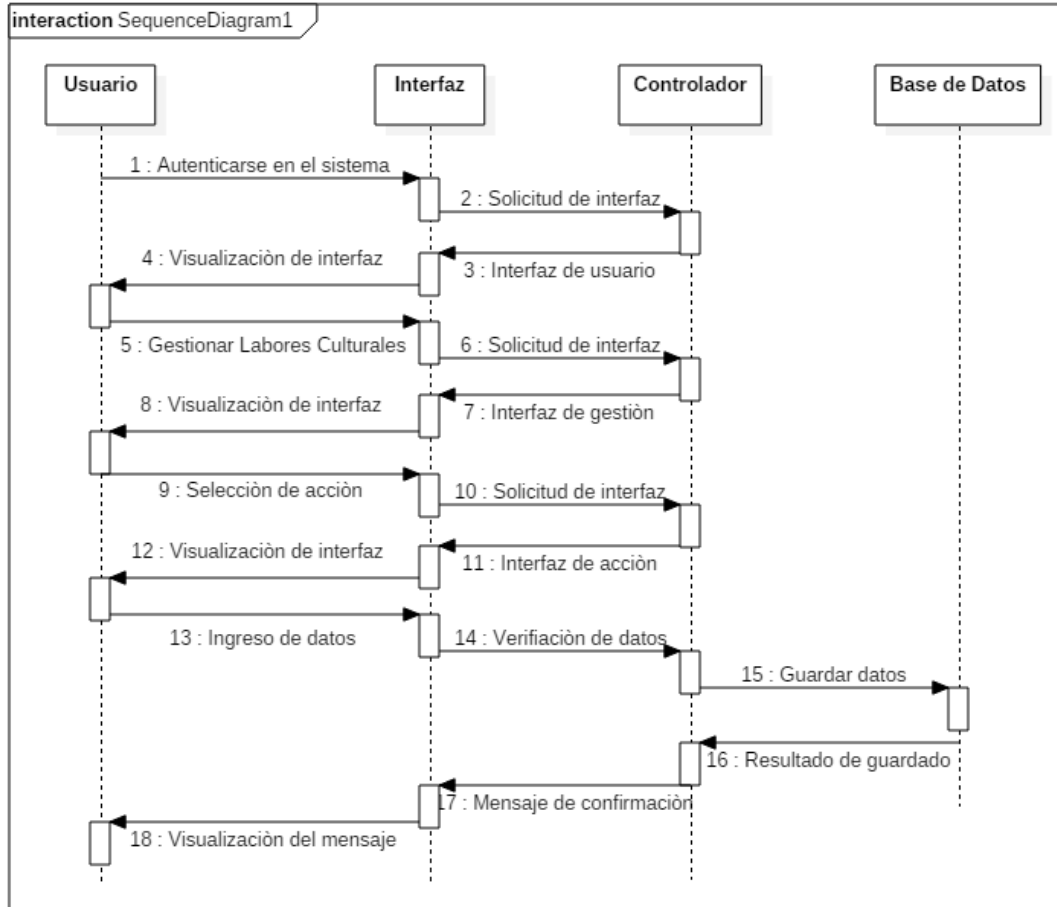
Descripción: Este diagrama nos muestra la secuencia de pasos a seguir para realizar la gestión de los lotes de una finca, el cual se encuentra responsable el perfil Dueño de Finca. En caso de que el usuario digite mal los datos en los formularios el sistema mostrara los errores y los solicitara nuevamente.

Ilustración 17 Diagrama de Secuencia #10 Gestión de Cultivos



Descripción: Este diagrama nos muestra la secuencia de pasos a seguir para realizar la gestión de cultivos correspondientes a una finca o lote, el cual se encuentra responsable el perfil Administrador. En caso de que el usuario digite mal los datos en los formularios el sistema mostrara los errores y los solicitara nuevamente.

Ilustración 18 Diagrama de Secuencia #11 Gestión de Labores Culturales



Descripción: Este diagrama nos muestra la secuencia de pasos a seguir para realizar la gestión de las labores culturales asociadas a un cultivo, el cual se encuentra responsable el perfil Administrador. En caso de que el usuario digite mal los datos en los formularios el sistema mostrara los errores y los solicitara nuevamente.

7.4 SPRINT 3

7.4.1 HISTORIAS DE USUARIO

Tabla 28 historia de Usuario 17

Historia de Usuario	
Número: 17	Nombre: Registro de producción
Prioridad en Negocio:	Iteración Asignada: 3
Descripción: Yo como cliente espero que el sistema permita al usuario administrador el registro de la producción de un cultivo en cada lote, este registro se debe realizar mediante los siguientes datos: <ul style="list-style-type: none">- Fecha- Lote(s) asociado(s)- Cultivo- Trabajador- Producción total por trabajador- Valor a pagar por kilogramo de café	
Observaciones:	

Tabla 29 Historia de Usuario 18

Historia de Usuario	
Número: 18	Nombre: Pago a trabajadores
Prioridad en Negocio:	Iteración Asignada: 3
Descripción: Yo como cliente espero que el sistema permita al usuario administrador la visualización del total a pagar por cada producción a cada trabajador. El sistema debe mostrar los pagos que se encuentren pendientes a trabajadores por labores culturales que no involucren un registro en producción.	
Observaciones:	

Tabla 30 Historia de Usuario 19

Historia de Usuario	
Número: 19	Nombre: Seguimiento de producción
Prioridad en Negocio:	Iteración Asignada: 3
Descripción: Yo como cliente espero que el sistema permita al usuario administrador la visualización de un reporte de producción, costos, precios y ganancias de cada producción o de un periodo	

de tiempo especificado

Observaciones:

El reporte puede ser visualizado mediante tablas o mediante gráficos de barras

7.4.2 VISTA DE ESCENARIOS (CASOS DE USO)

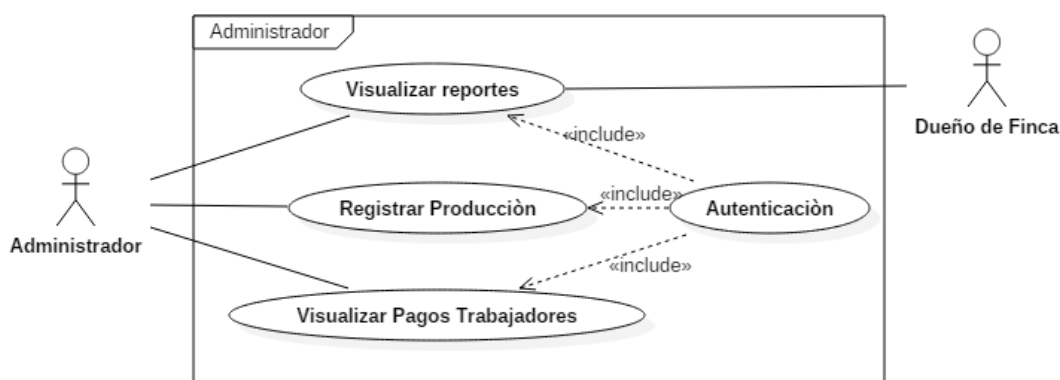


Ilustración 19 Diagrama Caso de Uso: Producción

Tabla 31 Caso de Uso 12

Caso de uso # 12	
Caso de uso	Registrar Producción
Actores	Administrador
Propósito	Permitir el registro de la producción de un lote
Resumen	Este caso de uso nos describe el curso de los eventos para el registro de producción de un lote
Prerrequisito	Caso de uso #1, #9, #4
Curso normal de los eventos	
Acción de los actores	Respuesta del sistema
1. El usuario se autentica en el sistema como perfil administrador	

2. Ingresar a la pestaña “Registro Producción”	3. El sistema despliega un formulario para el registro de la producción
4. El usuario digita los datos solicitados por el sistema y selecciona “Guardar”	5. El sistema verifica los datos y guarda los cambios.
Curso alterno	
Acción 2: Si el usuario ingresa datos erróneos, el sistema debe mostrar un mensaje con los datos incorrectos y regresar al paso 4	

Tabla 32 Caso de Uso 13

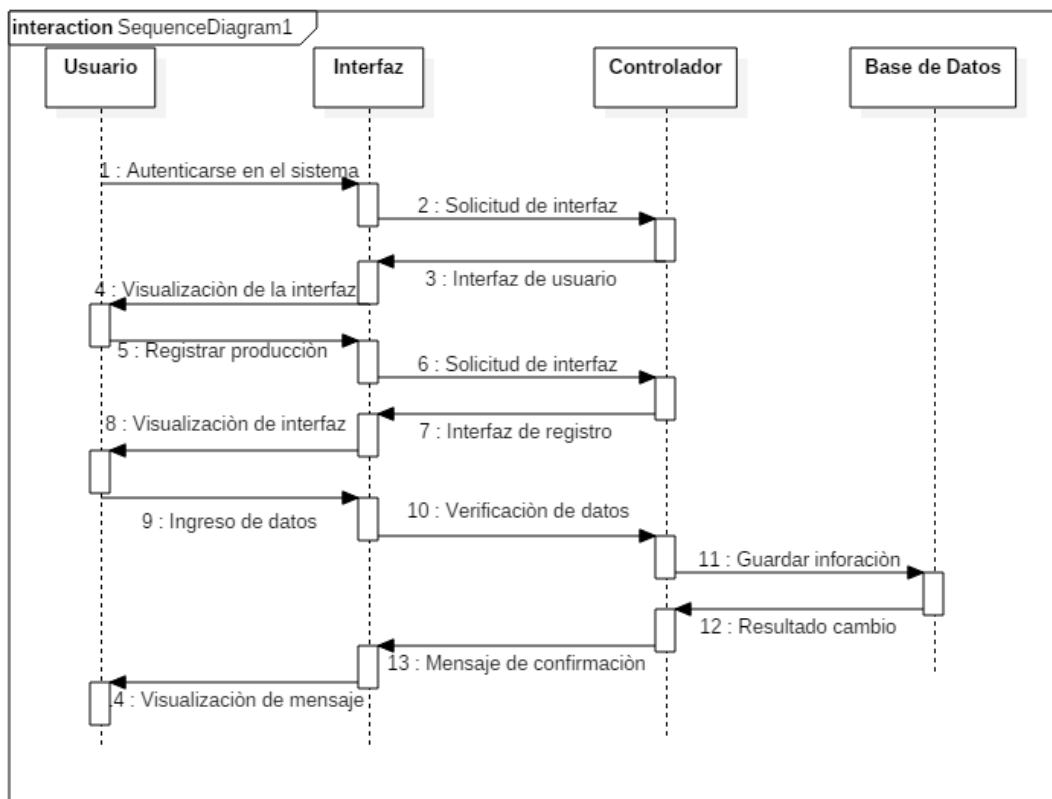
Caso de uso # 13	
Caso de uso	Visualizar reportes
Actores	Administrador, dueño de finca
Propósito	Permitir la visualización de reportes
Resumen	Este caso de uso nos describe el curso de los eventos para la visualización de reportes
Prerrequisito	Caso de uso #1, #12
Curso normal de los eventos	
Acción de los actores	Respuesta del sistema
1. El usuario se autentica en el sistema como perfil administrador o dueño de finca	
2. Ingresar a la pestaña “Reportes de Producción” y seleccionar la producción o el periodo de tiempo que desea conocer el reporte	3. El sistema despliega un formulario para la visualización del reporte

Tabla 33 Caso de Uso 14

Caso de uso # 14	
Caso de uso	Visualizar pagos
Actores	Administrador
Propósito	Permitir la visualización de pagos a trabajadores
Resumen	Este caso de uso nos describe el curso de los eventos para la visualización del pago a trabajadores
Prerrequisito	Caso de uso #1, #12
Curso normal de los eventos	
Acción de los actores	Respuesta del sistema
1. El usuario se autentica en el sistema como perfil administrador o dueño de finca	
2. Ingresar a la pestaña “Gestionar Trabajadores” y seleccionar “Pagos”	3. El sistema despliega un formulario con el total a pagar a los trabajadores
Curso alterno	

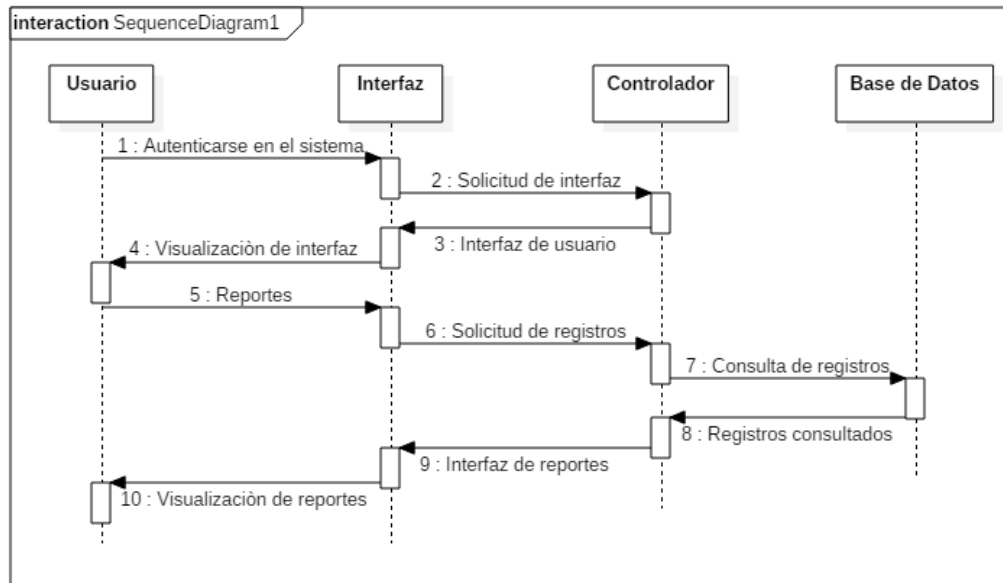
7.4.3 VISTA DE PROCESOS (DIAGRAMAS DE SECUENCIA)

Ilustración 20 Diagrama de Secuencia # 12 Registrar Producción



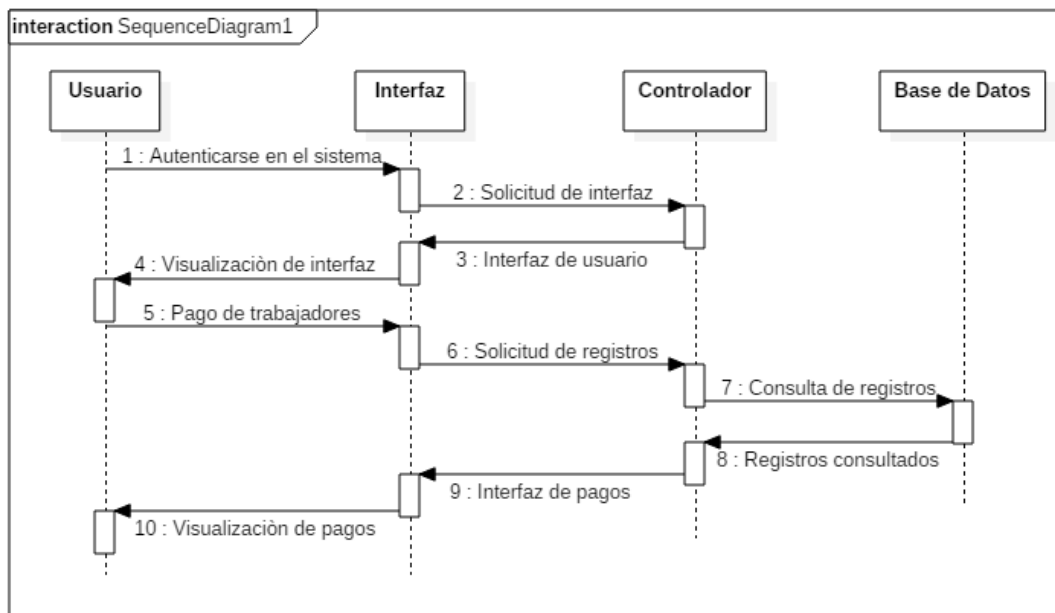
Descripción: Este diagrama nos muestra la secuencia de pasos a seguir para realizar el registro de la producción de un lote de la finca, el cual se encuentra como responsable el perfil Administrador. En caso de que el usuario digite mal los datos en los formularios el sistema mostrara los errores y los solicitara nuevamente.

Ilustración 21 Diagrama de Secuencia #13 Visualizar Reportes



Descripción: Este diagrama nos muestra la secuencia de pasos a seguir para realizar la visualización de reportes de producción, costos, gastos, ganancias y precios de un lote o de un cultivo, el cual se encuentra como responsable el perfil Administrador y perfil Dueño de Finca.

Ilustración 22 Diagrama de Secuencia #14 Visualizar Pagos



Descripción: Este diagrama nos muestra la secuencia de pasos a seguir para realizar la visualización de pagos a trabajadores que se encuentren pendientes, el cual se encuentra como responsable el perfil Administrador.

7.5 VISTA LÓGICA (DIAGRAMA DE CLASES)

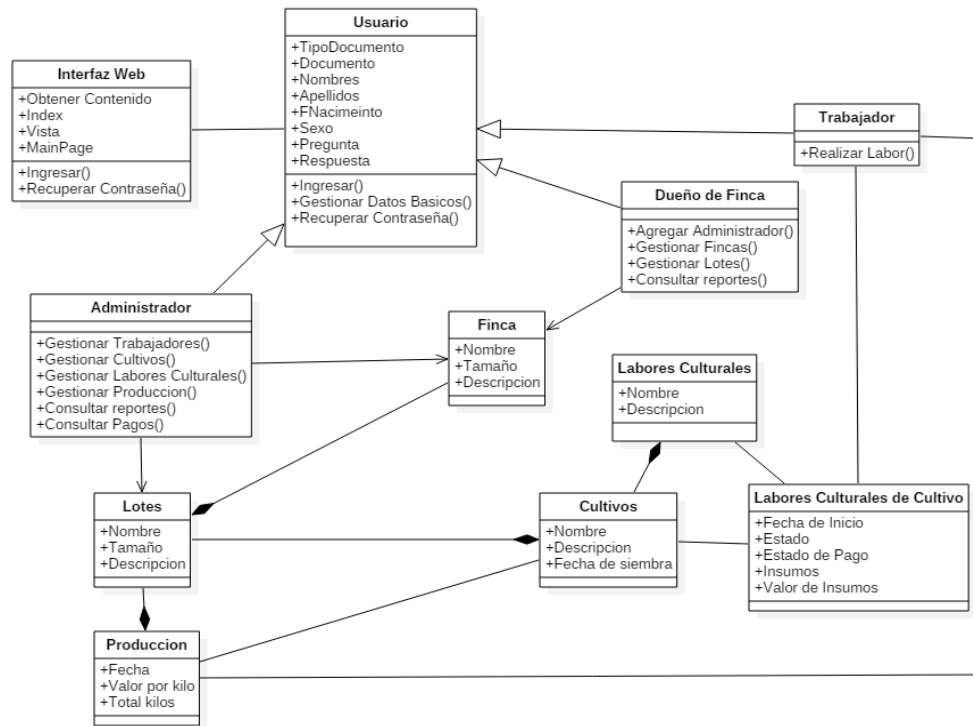


Ilustración 23 Vista Logica

7.6 VISTA DE DESARROLLO (DIAGRAMA DE COMPONENTES)

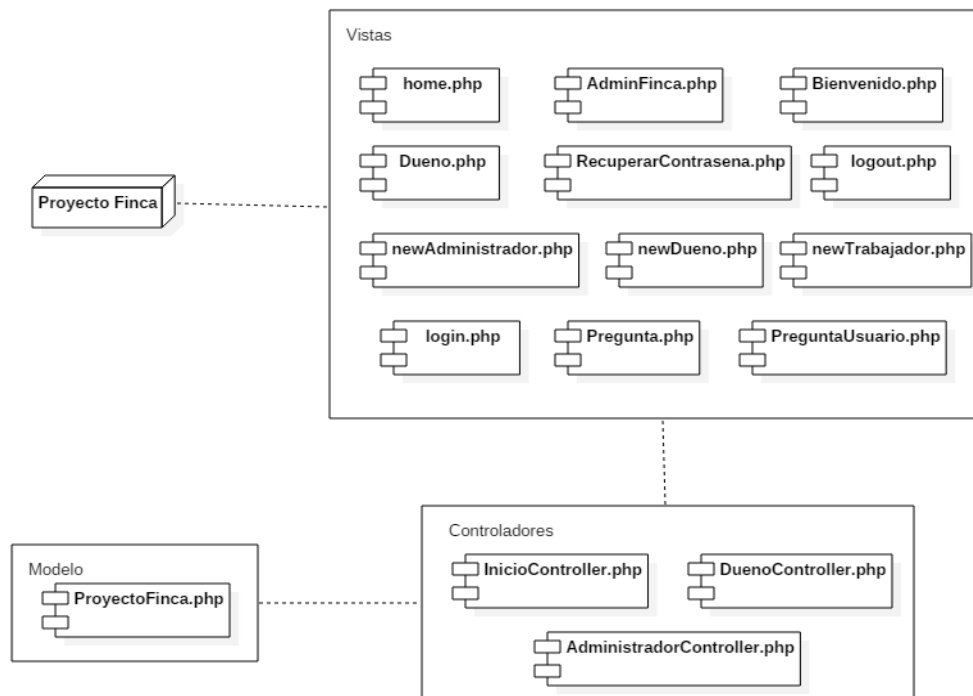


Ilustración 24 Vista de Desarrollo

7.8 VISTA FÍSICA (DIAGRAMA DE DESPLIEGUE)

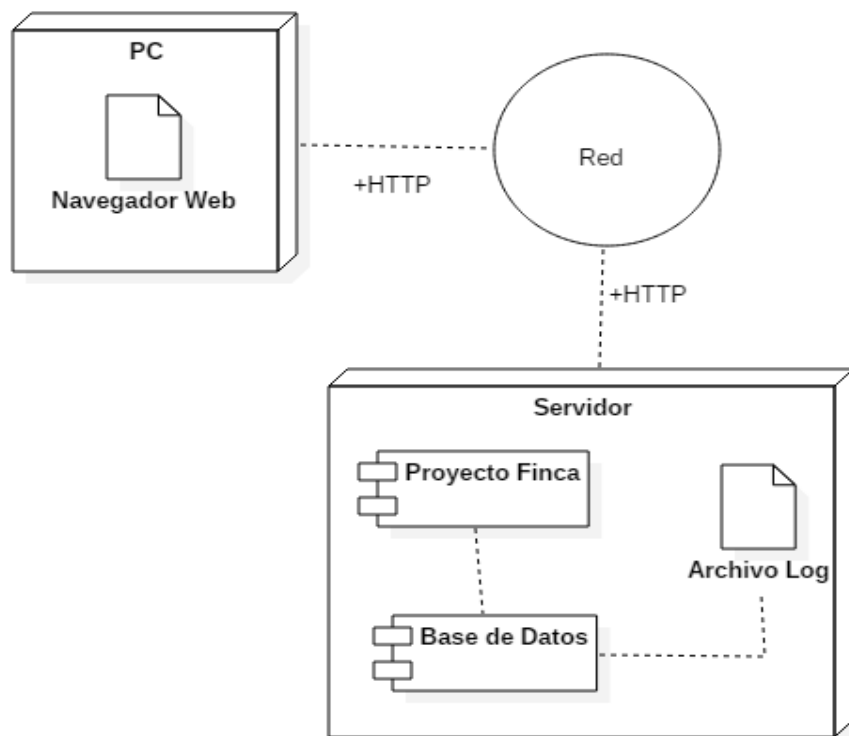


Ilustración 25 Vista Física

7.9 MODELO RELACIONAL BASE DE DATOS

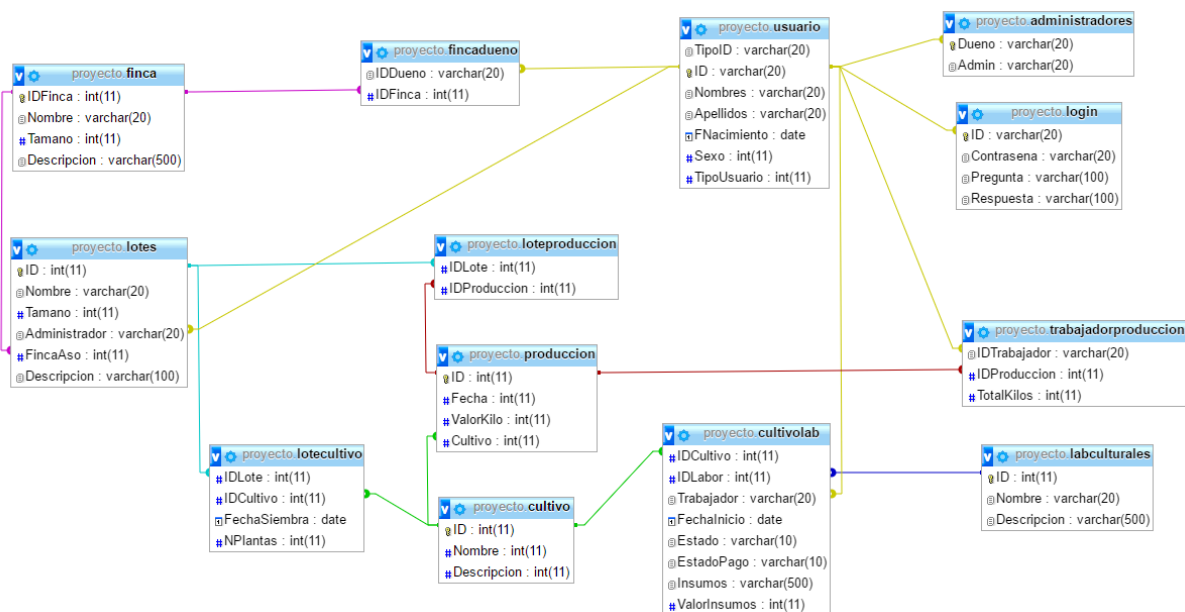


Ilustración 26 Base de Datos

8. APLICATIVO

A continuación, se muestran algunos pantallazos del aplicativo, como lo es el login, la página de inicio de los perfiles Dueño de Finca y Administrador

Bienvenido Sr(a) Rosalba Torres

Gestionar Datos Basicos

Gestionar Finca

Gestion Administrador

Gestion Contraseña

Salir

Nueva Finca

Administrar Mis Fincas

Nueva Finca

Nombre:

Tamaño:

Numero de lotes

Descripcion:

Bienvenido Sr(a) Hernando Antonio Torres

Gestionar Datos Basicos

Gestionar Trabajadores

Gestionar Lotes

Gestion Contraseña

Salir

Gestión de Datos Básicos

Tipo de Identificación

Numero Documento

Nombre

Apellido

Fecha de Nacimiento

Sexo

Aceptar

Bienvenido

Usuario

Contraseña:

Aceptar

Registrar Dueño

Olvido su Contraseña

9. CONCLUSIONES

Realizar el análisis y la especificación de requerimientos, de manera preliminar al desarrollo, permitió identificar y clasificar los requerimientos de mayor incidencia en el proyecto.

El análisis temprano de la arquitectura ayuda a reducir las probabilidades de fallo que pueden tener como consecuencia fracasos en los proyectos.

Las metodologías ágiles son útiles para el desarrollo rápido de software, teniendo ventajas enormes como entregar prototipos funcionales en poco tiempo, los cuales atraviesan el ciclo de vida del software y se encuentran debidamente documentados, ayudando a asegurar la calidad del producto.

La utilización de un framework, aporta ventajas de tiempo en el desarrollo puesto que el objeto principal de estos es no solo brindar una forma organizada de codificar, si no servir como estructura de trabajo y plantilla de codificación.

El diseño de software ofrece una vista global de lo que es el sistema y sus escenarios de uso, mostrándolo de manera lógica, dinámica y estructural.

La realización de pruebas de caja negra (pruebas funcionales), dentro de cada iteración, garantiza el cumplimiento de los requerimientos funcionales.

El correcto desarrollo de un producto de software debe atravesar y completar todas las etapas del ciclo de vida: análisis, diseño, codificación y pruebas.

La realización de un manual técnico permite que los usuarios puedan instalar el aplicativo en las máquinas correctas sin la necesidad de contar con un ingeniero.

Los módulos desarrollados en el aplicativo permiten a los dueños de fincas y administradores llevar un control sobre las actividades desarrolladas en el cultivo del café, agilizando el tiempo en tareas como el pago a trabajadores.

10. BIBLIOGRAFÍA

ALVAREZ, J. Ingeniería de software. Técnicas de, 2008.

APARICIO, Alexandra. Ingeniería de Software. EN: Datateca, Universidad Nacional Abierta y a Distancia, 2012.

FALGUERAS, Benet Campderrich. Ingeniería del software. Editorial UOC, 2002.

GONZÁLEZ, Yanette Díaz; ROMERO, Yenisleidy Fernández. Patrón Modelo-Vista-Controlador. Revista Telem@tica, 2012, vol. 11, no 1.

HUMPHREY, Watts S. A discipline for software engineering. Addison-Wesley Longman Publishing Co., Inc., 1995.

JACOBSON, Ivar; BOOCH, Grady; RUMBAUGH, James. El proceso unificado de desarrollo de software. Reading: Addison Wesley, 2000.

KNIBERG, Henrik. Scrum and XP from the Trenches: How we do Scrum. Lulu. com, 2007.

KRUCHTEN, Philippe. Planos Arquitectónicos: El Modelo de 4+ 1 Vistas de la Arquitectura del Software. IEEE Software, 1995, vol. 12.

SCHWABER, Ken. Scrum development process. In Business Object Design and Implementation. Springer London, 1997.

AGUDELO Rendón Paola Andrea, ALCALDE Guevara Denier Adolfo. Diseño e implementación de un prototipo de sistema de información en la web para las empresas prestadoras del servicio público de taxis en Pereira, Tesis, Facultad de Ingenierías Eléctrica, Física y Ciencias de la Computación, Programa de Ingeniería de Sistemas y Computación, Universidad Tecnológica de Pereira, 2015

CODEIGNITER. Guía de usuario. [Consultado el 15 marzo de 2016] :<
http://www.codeigniter.com/user_guide/>

ALISTAIR COCKBURN Metodologías Cristal [Consultado el 16 marzo de 2016] disponible en: <<http://alistair.cockburn.us/crystal/crystal.html>>

JAVA SCRIPT Documentación de java Script [Consultado el 18 marzo de 2016] disponible en: <<https://www.javascript.com/>>

ICONTEC Normas ICONTEC [Consultado el 18 de marzo de 2016] disponible en: <<http://normasicontec.org/>>

W3C Estándares [Consultado el 25 de marzo de 2016] disponible en: disponible en: <
<http://www.w3.org/>>

DOCUMENTACIÓN ÁGIL [Consultado el 25 de marzo de 2016] disponible en: <
http://www.literateprogramming.com/quotes_ad.html>

Manual Técnico

[Escriba el subtítulo del documento]

Daniel Torres Bedoya
09/06/2016

Contenido

1-	INSTRUCCIONES DE INSTALACIÓN DEL LOCALHOST	63
a.	Instalación.....	63
b.	Ejecución	68
c)	Reinicio.....	70
1-	REQUISITOS DEL SISTEMA.....	71
2-	INSTALACIÓN Y CONFIGURACIÓN DEL SISTEMA.....	72
a)	Instalación.....	72
b)	Configuración	72
3-	INSTALACIÓN Y CONFIGURACIÓN DE LA BASE DE DATOS.....	73
a)	Instalación.....	73
b)	Configuración	73
5-	COPIAS DE SEGURIDAD	74
6-	DEFINICIÓN DE IMPLEMENTACIÓN O LIBRERÍAS UTILIZADAS.....	75
7-	GLOSARIO.....	76

1. INSTRUCCIONES DE INSTALACIÓN DEL LOCALHOST

1.1 INSTALACIÓN

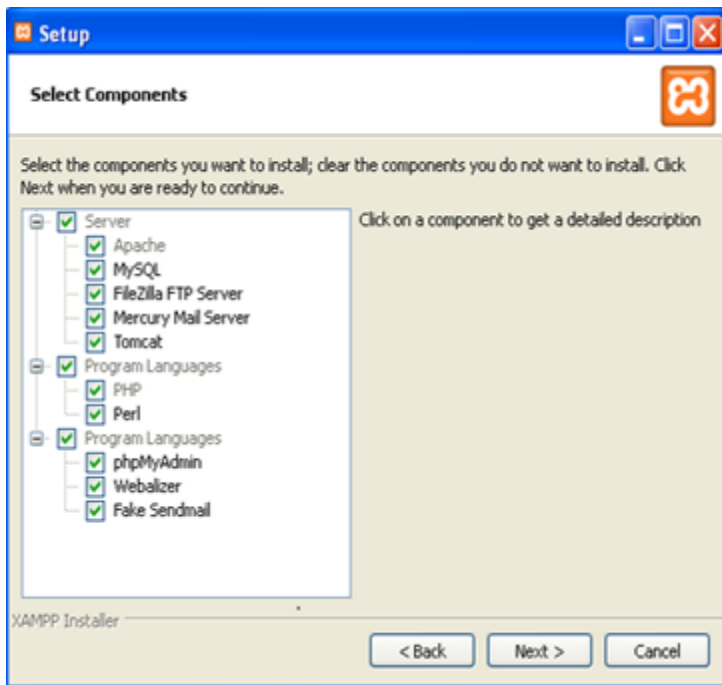
Para la elección de un servidor local se tienen dos opciones Xampp y Wamp, los cuales se encuentran de forma gratuita en internet. Al momento de instalar el sistema de información en el servidor local se debe tener en cuenta que los archivos deben incluirse en un directorio específico de acuerdo a la elección del servidor, en el caso de Xampp se incluye los archivos en la carpeta “htdocs”, y “www” en el caso de Wampp.

Los pasos para la instalación del servidor local Xampp en el sistema operativo Windows son:

- 1- Acceder a la página web: www.apachefriends.org/es/index.html y descargar el ejecutable correspondiente a nuestro sistema operativo.



- 2- Abrir la carpeta de descargas y se busca el ejecutable descargado.



3- Iniciar el proceso de instalación abriendo el ejecutable, seguido se abrirá el panel de instalación en donde se debe seleccionar los componentes a instalar.

4- Se elige la carpeta en la que se instalará el servidor local o se deja la ubicación por defecto y se da click en “Next”.

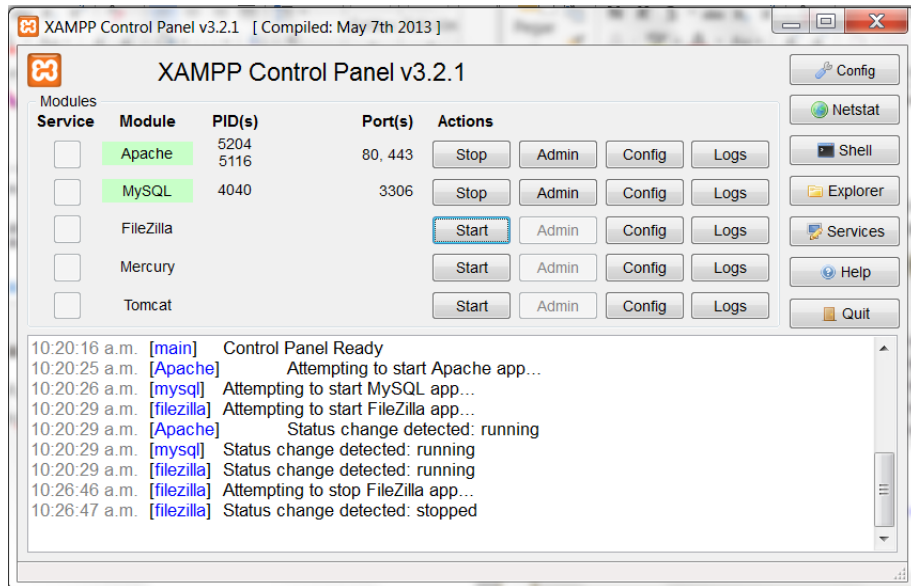
5- Se informa de las aplicaciones opcionales, se da click en “Next”.



6- Finalizada la instalación se solicita iniciar el panel de control y se da click en “Finish”.

7- Al aceptar la opción de iniciar el panel de control se solicitan los permisos para la inclusión de la Apache y MySQL en el firewall de Windows, se da click en permitir acceso.

8- Una vez en el panel de control se presiona el botón “Admin” para iniciar la configuración de idioma de Xampp y ver el panel de control en el navegador.

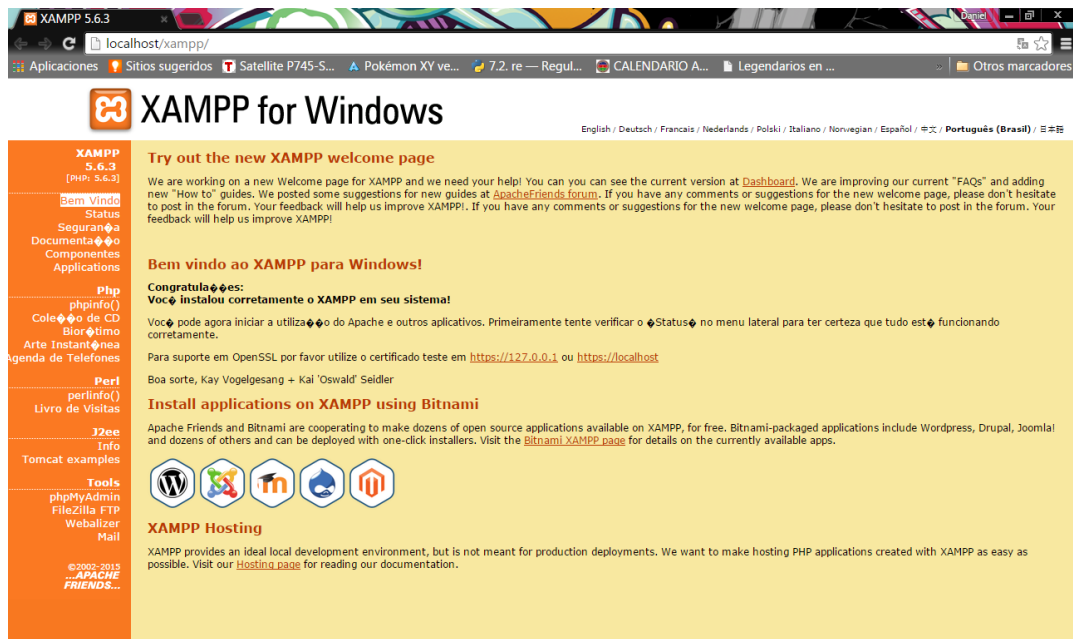


9- Se elige la opción de idioma que se desea.

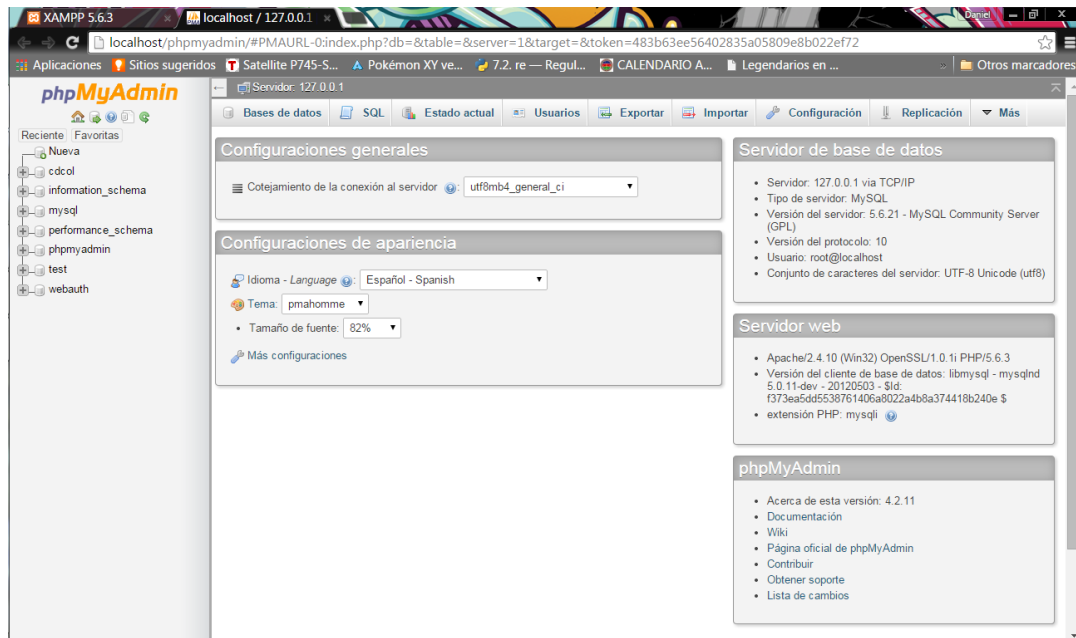


[English](#) / [Deutsch](#) / [Francais](#) / [Nederlands](#) / [Polski](#) / [Italiano](#) / [Norwegian](#) / [Español](#) / [中文](#) / [Português \(Brasil\)](#) / [日本語](#)

10- Y por último se abre el panel de navegación de Xampp.



11- De igual manera se da click en “Admin” para acceder al panel de navegación gráfico de MySQL phpMyAdmin.

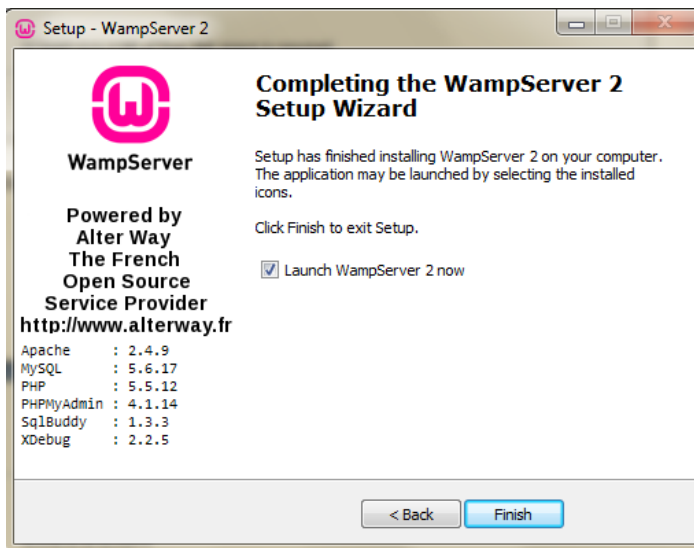
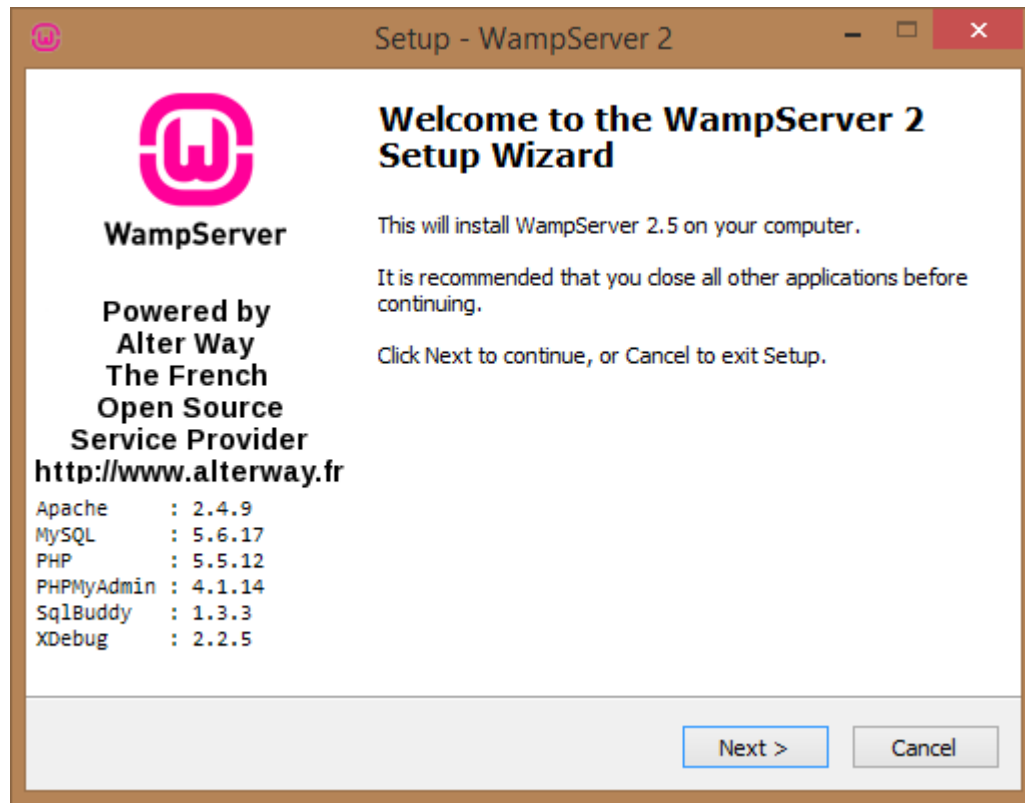


Los pasos para la instalación del servidor local Wamp en el sistema operativo Windows son:

- 1- Acceder a la página www.wampserver.com/en/ e ingresar a la pestaña “Download” y escoger la versión adecuada para el tipo de máquina.



- 2- Acceder a la carpeta de descargas y buscar el ejecutable descargado.
- 3- Después de ejecutarlo se mostrará el panel de configuración del instalador.



- 4- Se hace click en el botón “Next” y aceptar los términos y condiciones, posterior el software pide especificar la carpeta en que se instalará el servidor local.
- 5- Al momento de terminar la instalación se da click en “Finish” para terminar.

1.2 EJECUCIÓN

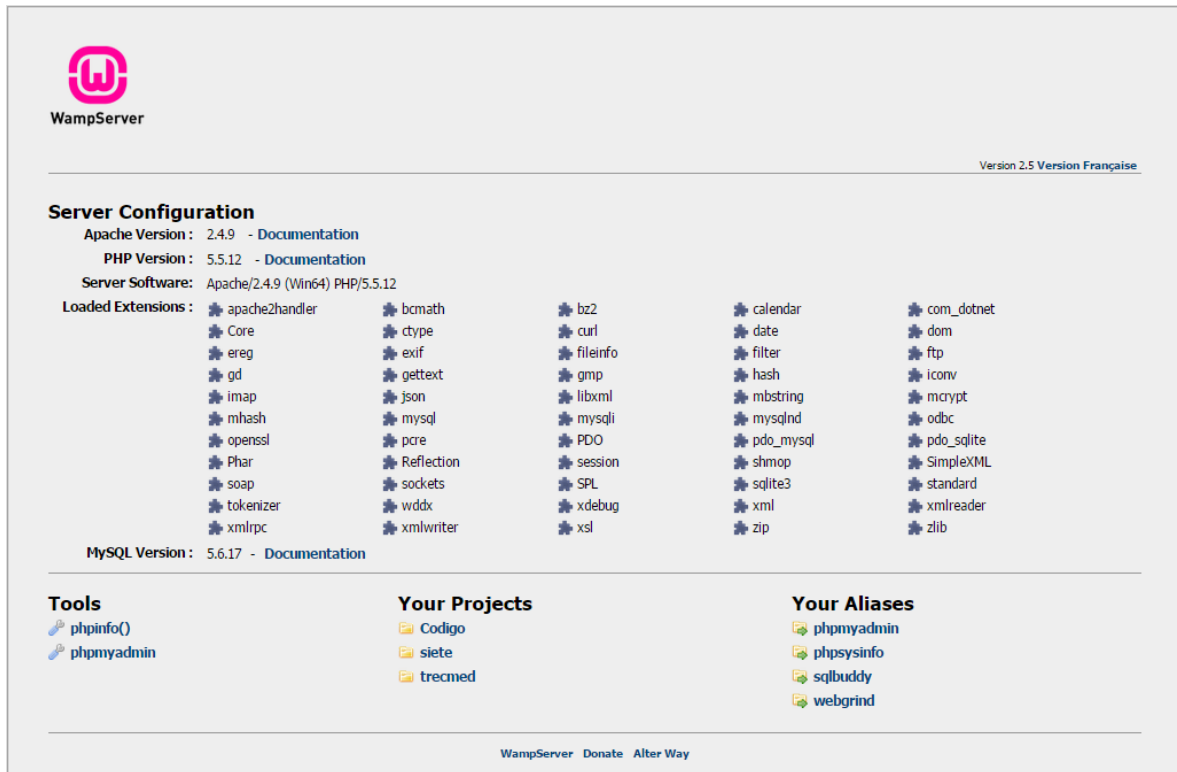
Xampp y Wampp utilizan los puertos 80 HTTP (“*HyperText Transfer Protocol*”, Protocolo de Transferencia de HiperTexto), WWW; y 443 (HTTPS/SSL usado para la transferencia segura de páginas web). Estos puertos suelen causar problemas si existen otras aplicaciones que los utilizan con propósitos de consumo de ancho de banda, por ejemplo, TeamViewer y Skype. De ser así se deben configurar los puertos mencionados y usar los puertos 8080 y 4430.

Una vez terminada la instalación del servidor local, se comprueba que Apache está corriendo se debe escribir en el navegador la URL “localhost” o la dirección IP 127.0.0.1 y se podrá ver la interfaz de Xampp o Wampp.

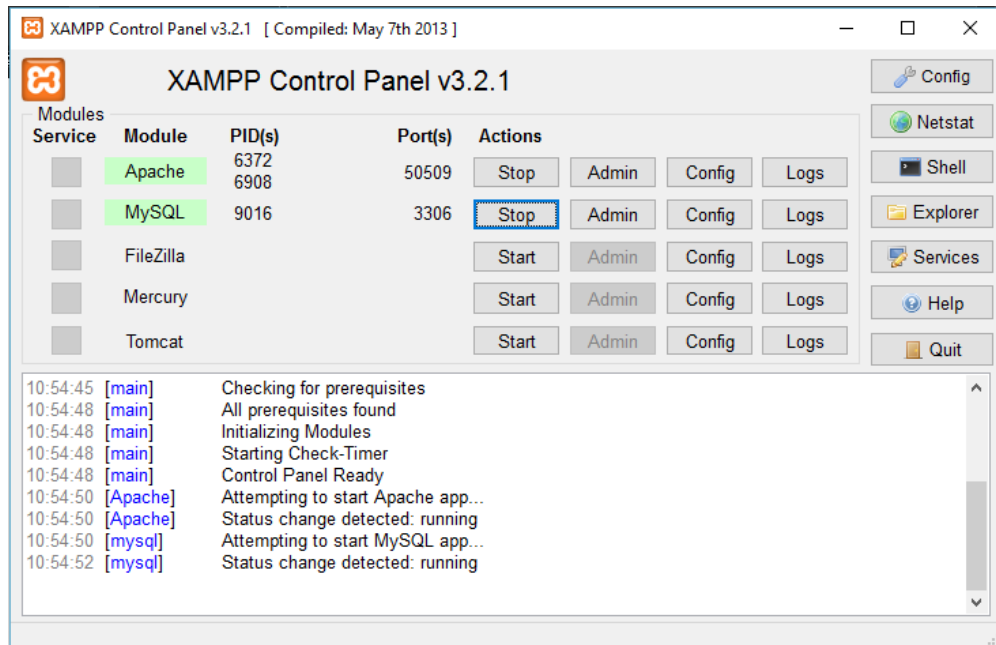
En esta interfaz se podrá acceder a phpMyAdmin versión 4.4.14 que es la consola o aplicación para correr MySQL de forma gráfica.

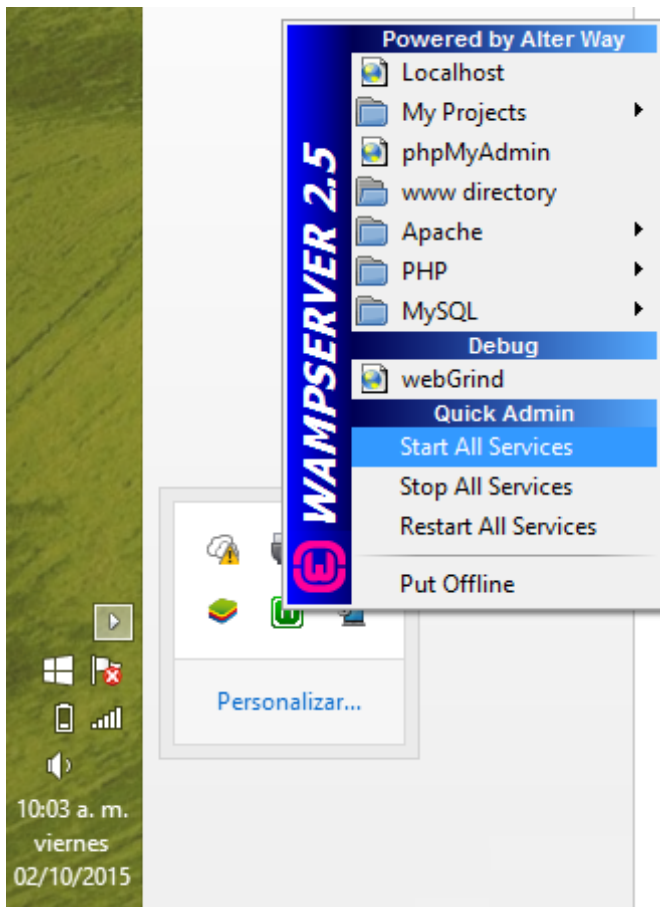
Hechas las pruebas satisfactorias, se tendrá disponible el servidor local.





Panel de Control de Xampp: en el panel de control de Xampp se inician los servicios Apache y MySQL.





Para activar los servicios tales como Apache y MySQL, se busca el icono de WampServer en la barra de tareas del computador, se hace click izquierdo y se elige la opción de iniciar todos los servicios (*“Start All Services”*) para activarlos.

1.3 REINICIO

En caso de presentarse un problema en el sistema operativo o un bloqueo se debe realizar un *“reset”* de todos los servicios tanto de la base de datos como el servidor, basta con hacer click en *“Stop”* y luego en *“Start”* (la opción *“Restart”* disponible en Wampp).

2. REQUISITOS DEL SISTEMA

Para PHP:

- Windows Vista Service Pack 2 o posterior.
- Windows Server 2008 Service Pack 2 o posterior.
- Windows Server 2008 R2.
- Windows 7.
- Windows 8.
- Windows 10.

Para Servidor Local:

- Procesador Intel pentium4, Intel Centrino, Intel Xeon, o Intel Core Duo (o compatible).
- Microsoft Windows XP con service pack 2, Windows vista home Pentium, business, ultimate o enterprise (Para 32 bits) o versiones posteriores.
- 512 MB de RAM.
- 40 MB de espacio disponible en disco duro.
- Conexión a internet.

Para el sistema:

- Apache 2.4.12
- PHP 5.6.6
- MySQL 5.7.4

3. INSTALACIÓN Y CONFIGURACIÓN DEL SISTEMA

3.1 INSTALACIÓN

Para la instalación del sistema en un servidor siga el procedimiento habitual, en caso de usar un servidor local copie la carpeta Siete y ubíquela en la carpeta correspondiente al Servidor utilizado (ver 1-INSTRUCCIONES DE INSTALACIÓN DEL LOCALHOST sección a) Instalación).

3.2 CONFIGURACIÓN

Para la configuración del sistema (URL de acceso) siga los siguientes pasos:

- 1- Diríjase a la ubicación de la carpeta ProyectoFinca.
- 2- Ingrese al siguiente fichero: “...ProyectoFinca/application/config”
- 3- Abra el archivo “config.php”.
- 4- Ubique la línea y verifique que la configuración se encuentre como en la siguiente imagen.

```
13 |  
14 | If this is not set then CodeIgniter will try guess the protocol, domain  
15 | and path to your installation. However, you should always configure this  
16 | explicitly and never rely on auto-guessing, especially in production  
17 | environments.  
18 |  
19 | */  
20 | $config['base_url'] = 'http://localhost/ProyectoFinca/';|  
21 |
```

NOTA: Por favor no abra, ni modifique ningún otro archivo o campos diferentes a los mencionados en los pasos, y evite modificar el archivo anteriormente mencionado una vez el sistema esté en uso, cualquier manipulación indebida de los archivos podría poner en riesgo la integridad del sistema y los datos que este contiene

4. INSTALACIÓN Y CONFIGURACIÓN DE LA BASE DE DATOS

4.1 INSTALACIÓN

Para la instalación de la base de datos en un servidor siga el procedimiento habitual usando el Script de la base de datos ProyectoFinca.

En caso de usar un servidor local se deben seguir los siguientes pasos:

- 1- Inicie sesión en phpMyAdmin: "localhost/phpmyadmin".
- 2- Cree una base de datos de nombre Proyecto.
- 3- Seleccione la opción Importar seguido seleccione el script de la base de datos, haga click en Continuar.

4.2 CONFIGURACIÓN

Para la configuración de la base de datos siga los siguientes pasos:

- 1- Diríjase a la ubicación de la carpeta siete.
- 2- Ingrese al siguiente fichero: *"...ProeyctoFinca/application/config"*
- 3- Abra el archivo *"database.php"*.
- 4- Ubique la línea 62 y escriba los datos correspondientes a la base de datos, tales como nombre del servidor, nombre de usuario, contraseña y nombre de la base de datos.

```
62 $active_group = 'default';
63 $query_builder = TRUE;
64
65 $db['default'] = array(
66     'dsn' => '',
67     'hostname' => 'localhost',
68     'username' => 'root',
69     'password' => '',
70     'database' => 'proyecto',
71     'dbdriver' => 'mysqli',
72     'dbprefix' => '',
73     'pconnect' => TRUE,
74     'db_debug' => TRUE,
75     'cache_on' => FALSE,
76     'cachedir' => '',
77     'char_set' => 'utf8',
78     'dbcollat' => 'utf8_general_ci',
79     'swap_pre' => '',
80     'encrypt' => FALSE,
81     'compress' => FALSE,
82     'stricton' => FALSE,
83     'failover' => array(),
84     'save_queries' => TRUE
85 );
86
```

NOTA: Por favor no abra, ni modifique ningún otro archivo o campos diferentes a los mencionados en los pasos, y evite modificar el archivo anteriormente mencionado una vez el sistema esté en uso, cualquier manipulación indebida de los archivos podría poner en riesgo la integridad del sistema y los datos que este contiene

5. COPIAS DE SEGURIDAD

Para realizar los procesos de respaldo de información en un servidor, se deben seguir los pasos establecidos en las políticas de seguridad de información de la empresa en el anexo o apartado referido a copias de seguridad.

En caso de usar un servidor local se deben seguir los siguientes pasos:

- 1- Inicie sesión en phpMyAdmin: "localhost/phpmyadmin".
- 2- Seleccione la base de datos en la que desea realizar la copia de seguridad.
(Se mostrarán las propiedades de la base de datos).
- 3- Haga clic en Exportar. Esto generará un archivo de copia de seguridad que se puede utilizar para restaurar la base de datos (Asegúrese que la opción datos o estructura y datos este seleccionada).
- 4- Puede cambiar el nombre del archivo de copia de seguridad y la compresión seleccionando la opción Personalizado.
- 5- Dé por finalizado el proceso haciendo clic en Continuar. Esto puede tardar unos minutos en completarse.
- 6- En caso de restauración seleccione la opción *"Import"* seguido seleccione el archivo creado anteriormente y haga click en Continuar.

6. DEFINICIÓN DE IMPLEMENTACIÓN O LIBRERÍAS UTILIZADAS

- Jscal2: también conocido como Dynarch Calendar, es una librería escrita en JavaScript que usa estilos css, usada para la generación de los calendarios en los diferentes formularios del sistema²⁷.
- Fpdf17: es una clase escrita en PHP que permite generar documentos PDF directamente desde PHP²⁸.

²⁷ Documentación de Jscal2 disponible en: <http://www.dynarch.com/jscal/>

²⁸ Librería Fpdf disponible en <http://www.fpdf.org/>

7. GLOSARIO

- **Ejecutable:** Es un archivo que tiene la capacidad de poder ser ejecutado de forma independiente o que no necesita ser ejecutado por una aplicación externa.
- **Firewall:** Es un software o hardware que comprueba la información procedente de Internet o de una red y, a continuación, bloquea o permite el paso de ésta al equipo.
- **Sistema operativo:** Es el software básico de una computadora que provee una interfaz entre el resto de programas del ordenador, los dispositivos hardware y el usuario.
- **Reset:** Reponer o reiniciar un sistema.
- **Servidor:** Es un ordenador remoto que provee los datos solicitados por parte de los navegadores de otras computadoras.
- **Base de datos:** Es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.
- **Navegador:** Es un software que permite el acceso a Internet, interpretando la información de distintos tipos de archivos y sitios web para que éstos puedan ser visualizados.
- **Librería:** Es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.
- **Programa:** Es una secuencia de instrucciones, escritas para realizar una tarea específica en una computadora.

PRODUCT BACKLOG

HU	ATRIBUTOS DEL SISTEMA QUE DEBEN SER CONSIDERADOS EN TODOS LOS SPRINT
1	Navegadores Debe ser un software de arquitectura web y los usuarios deben poder acceder a él por Internet, usando navegadores Internet Explorer 10 o superior, Firefox 23 o superior, Google Chrome 28.0 o superior.
2	Interfaz de usuario Cumplir con requisitos mínimos de usabilidad de la W3C
3	Desempeño Cada petición que realice el usuario debe tener una respuesta del sistema en máximo 3 segundos y debe soportar al menos 10000 registros en la base de datos
SPRINT 1	
4	Tipos de usuario El sistema debe permitir la creación de los perfiles: dueño de finca, administrador y trabajadores.
5	Perfil dueño de finca El sistema debe permitir al usuario dueño el registro y edición de la siguiente información básica: <ul style="list-style-type: none"> * Tipo de identificación * Numero de identificación * Nombres * Apellidos * Fecha de nacimiento * Edad (La cual debe ser automática mediante el ingreso de la fecha de nacimiento) * Sexo
6	Perfil Administrador El sistema debe permitir al usuario dueño de finca la creación de administradores con la siguiente información básica: <ul style="list-style-type: none"> * Tipo de identificación * Numero de identificación * Nombres * Apellidos * Fecha de nacimiento * Edad (La cual debe ser automática mediante el ingreso de la fecha de nacimiento) * Sexo
7	Perfil Trabajador

	<p>El sistema debe permitir al usuario administrador la creación de usuarios trabajadores con la siguiente información básica:</p> <ul style="list-style-type: none"> * Tipo de identificación * Numero de identificación * Nombres * Apellidos * Fecha de nacimiento * Edad (La cual debe ser automática mediante el ingreso de la fecha de nacimiento) * Sexo <p>NOTA: Los únicos datos obligatorios para este perfil son: Nombre, Apellido y Sexo</p>
8	<p>Funciones perfil dueño de finca</p> <p>El perfil dueño de finca tenga como funciones:</p> <ul style="list-style-type: none"> * Gestionar sus datos básicos * Crear usuario administrador * Eliminar administrador * Recuperar contraseña
9	<p>Perfil administrador</p> <p>El perfil administrador tendrá como funciones:</p> <ul style="list-style-type: none"> * Gestionar sus datos básicos * Crear trabajadores * Gestionar información básica de trabajadores * Eliminar trabajadores * Recuperar contraseña
10	<p>Ingreso de usuarios</p> <p>El sistema debe permitir a los usuarios dueño de finca y administrador el ingreso al sistema mediante un usuario y una contraseña</p> <p>NOTA:</p> <ul style="list-style-type: none"> * Los usuarios trabajadores no tendrá acceso directo al aplicativo su información será manejada únicamente por los perfiles administrador. * Un usuario puede ser dueño de finca y administrador, en caso de que ocurra al momento de iniciar sesión se debe presentar la opción de ingresar a uno de los dos perfiles.
11	<p>Recuperación de contraseñas</p> <p>La recuperación de contraseñas para el perfil dueño de finca y el perfil administrador se realizará mediante una pregunta secreta que el usuario digito la primera vez que se logueo en el sistema y podrá modificar en la sección de datos básicos</p>
12	<p>Primer inicio de sesión</p> <p>Después de registrado el perfil dueño de finca y administrador estos podrán acceder al sistema con su número de identificación como contraseña, después de que inicien sesión por primera vez se debe desplegar un mensaje el cual permita el cambio de su contraseña y el registro de una pregunta secreta con fines de recuperación de contraseña</p>
SPRINT 2	
13	Gestión de finca

	<p>El sistema debe permitir al usuario dueño de finca la creación y edición de fincas con los siguientes datos básicos:</p> <ul style="list-style-type: none"> * Nombre * Tamaño (Numero de hectáreas) * Numero de lotes * Descripción
14	<p>Gestión de lotes</p> <p>El sistema debe permitir al usuario dueño de finca registrar y editar a cada lote los siguientes datos básicos:</p> <ul style="list-style-type: none"> * Nombre del lote * Tamaño del lote (En hectáreas) * Administrador asociado. * Finca asociada * Cultivos sembrados * Descripción del lote <p>NOTA: El numero sumado de hectáreas de cada uno de los lotes debe ser igual al tamaño total de la finca</p>
15	<p>Eliminación de finca</p> <p>El sistema debe permitir al usuario dueño de finca la eliminación de la finca. Cuando se elimine la finca no se deben eliminar la lista de administradores y trabajadores asociados a esta.</p>
16	<p>Eliminación de lotes</p> <p>El sistema debe permitir al usuario dueño de finca la eliminación de lotes asociados a una finca, el tamaño correspondiente del lote a eliminar debe ser adjudicado a otro(s) lote(s). Cuando se elimine el lote no se deben eliminar el administrador y los trabajadores asociados a este.</p>
17	<p>Registro de cultivos</p> <p>El sistema debe permitir al usuario administrador la creación de cultivos en cada lote, para cada cultivo se debe guardar la siguiente información:</p> <ul style="list-style-type: none"> * Nombre * Numero de platas * Edad * Labores culturales (Actividades de mantenimiento del cultivo) * Producción
18	<p>Registro labores culturales</p> <p>El sistema debe permitir al perfil administrador la creación y edición de la siguiente información de las labores culturales</p> <ul style="list-style-type: none"> * Nombre * Responsable * Numero de días a trabajar * Fecha de finalización * Estado (Pendiente, En progreso, Finalizada) * Herramientas o insumos necesarios * Valor de herramientas o insumos * Valor realización de la labor * Estado de pago (Pendiente, Cancelado)
19	<p>Administración de lotes</p>

	<p>El sistema debe permitir al usuario administrador la administración de los lotes de la siguiente forma:</p> <ul style="list-style-type: none"> * Asignar a cada lote cultivos * Asignar a cada cultivo labores culturales * Asignar a cada labor cultural un trabajador responsable
SPRINT 3	
20	Registro de producción
	<p>El sistema debe permitir al usuario administrador el registro de la producción de un cultivo en un lote(s) específico(s) registrando la siguiente información:</p> <ul style="list-style-type: none"> * Fecha * Lote(s) asociado(s) * Cultivo * Trabajador * Total producción por trabajador (Kilogramo) * Valor a pagar por kilogramo * Precio de venta por kilogramo
21	Pago a trabajadores
	<p>El sistema debe permitir al usuario administrador visualizar el total a pagar por cada trabajador en cada una de las producciones o periodos de tiempo</p> <p>El sistema debe mostrar los pagos que se encuentren pendientes a trabajadores por labores culturales que no involucren un registro en producción.</p>
22	Seguimiento de producción
	<p>El sistema debe permitir al perfil administrador realizar un seguimiento a la producción y análisis de los cultivos registrando los siguientes datos:</p> <ul style="list-style-type: none"> * Valor de cada kilogramo vendido * Valor total de la venta
23	Visualización producción
	<p>El sistema debe permitir a los perfiles dueño de finca y administrador la visualización de un reporte de costos, producción y ganancias asociadas a una producción y venta realizada o por periodo de tiempo.</p>
24	Gráficos de rendimiento
	<p>El sistema debe permitir a los perfiles dueño de finca y administrador la visualización de graficas de rendimiento de los cultivos pertenecientes a cada lote, en donde se muestre información de producción, costos y ganancias por periodos de tiempo especificados</p>